

# ĐÁP ÁN ĐỀ THI CUỐI KỲ

## I. Bảng phân bố câu hỏi

Đề 1	Đề 2	Đề 3	Đề 4	CDR (theo đề 1)
1	12	9	7	G1
2	20	18	10	G1
3	5	8	5	G1
4	1	11	17	G1
5	3	17	14	G1
6	10	1	12	G2, G4
7	17	6	16	G2
8	13	3	11	G1
9	16	7	19	G1
10	6	16	6	G2
11	2	4	3	G1
12	15	14	9	G2
13	8	15	20	G1
14	11	13	1	G2
15	4	12	13	G1
16	18	10	2	G1
17	19	20	18	G2, G4
18	14	5	15	G1
19	9	2	4	G2
20	7	19	8	G2

## II. Đáp án

## ĐỀ 1

Phần Trắc nghiệm (mỗi câu đúng được 0.3đ)

1B    2C    3B    4B    5C    6D    7C    8A    9B    10C  
11B   12B   13B   14A   15B   16C   17C   18B   19B   20A

Phần Tự luận:

Câu 1 (1đ):

Khai báo và khởi tạo các semaphore: (Trừ 0.25 đ nếu không khai báo và khởi tạo các semaphore sử dụng)

init(sem1,0); //khởi tạo semaphore sem1 có giá trị bằng 0

init(sem2,0); //khởi tạo semaphore sem2 có giá trị bằng 0

<pre>void T1(void) {  //T1 thực thi  signal(sem1) signal(sem1) }</pre>	<pre>void T2(void) {  wait(sem1)  //T2 thực thi  signal(sem2)  }</pre>	<pre>void T3(void) {  wait(sem1)  //T3 thực thi  signal(sem2)  }</pre>	<pre>void T4(void) {  wait(sem2) wait(sem2)  //T4 thực thi  }</pre>
------------------------------------------------------------------------	------------------------------------------------------------------------	------------------------------------------------------------------------	---------------------------------------------------------------------

Đặt đúng vị trí các hàm wait() và signal() vào các hàm thực thi bên trên được 0.25 đ/hàm thực thi.

Xét trên toàn bộ 4 hàm thực thi, nếu sinh viên đặt đúng ít hơn 3 vị trí thì được 0.25đ khuyến khích, từ 3-5 vị trí thì được 0.5đ, từ 6-7 vị trí thì được 0.75đ.

Nếu sinh viên sử dụng nhiều hơn 2 semaphore nhưng đáp ứng được yêu cầu đề bài thì vẫn được trọn điểm.

Câu 2 (1đ):

Request P4 (2, 3, 1, 3)  $\leq$  Need P4 (2, 3, 1, 5)

Request P4 (2, 3, 1, 3)  $\leq$  Available (3, 4, 4, 3)

(Trừ 0.25đ nếu sinh viên không xét hai điều kiện này)

Giả sử hệ thống đáp ứng yêu cầu thêm tài nguyên (2, 3, 1, 3) của tiến trình P4.

Trạng thái của hệ thống sau khi đáp ứng:

	Allocation				Max				Need				Available (Work)				
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	
P1	1	2	2	3	2	3	4	3	1	1	2	0	1	1	3	0	P1
P2	3	1	3	1	3	8	6	1	0	7	3	0	2	3	5	3	P4
P3	2	1	4	5	7	7	5	7	5	6	1	2	7	7	11	8	P5
P4	5	4	6	5	5	4	6	7	0	0	0	2	8	11	15	10	P2
P5	1	4	4	2	1	6	7	3	0	2	3	1	11	12	18	11	P3

Sau khi cấp phát cho P4 thì hệ thống vẫn có chuỗi an toàn <P1, P4, P5, P2, P3> cho nên hệ thống đáp ứng yêu cầu cấp phát cho P4.

- Tính đúng ma trận Need được 0.5 đ. Tính sai Need thì được 0.25đ và không chấm tiếp.
- Tính sai ma trận Available thì được 0.25đ.
- Tính đúng ma trận Available, xác định đúng chuỗi an toàn và kết luận đúng (cấp phát được) thì được 0.5đ.
- Tính đúng ma trận Available, nhưng xác định sai chuỗi an toàn hoặc kết luận sai (không cấp phát được) hoặc không kết luận gì thì được 0.25đ.

Câu 3 (2đ):

a. Giải thuật OPT có số lỗi trang là: 8

3	2	4	6	1	2	3	5	4	1	2	2	5	6	4	1	6	5	4	3
3	3	3	3	3	3	3	5	5	5	5	5	5	5	5	5	5	5	5	3
	2	2	2	2	2	2	2	2	2	2	2	2	6	6	6	6	6	6	6
		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
			6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
*	*	*	*	*			*						*						*

b. Giải thuật LRU có số lỗi trang là: 14

3	2	4	6	1	2	3	5	4	1	2	2	5	6	4	1	6	5	4	3
3	3	3	3	1	1	1	1	4	4	4	4	4	6	6	6	6	6	6	6
	2	2	2	2	2	2	2	2	1	1	1	1	1	4	4	4	4	4	4
		4	4	4	4	3	3	3	3	2	2	2	2	2	1	1	1	1	3
			6	6	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5
*	*	*	*	*		*	*	*	*	*			*	*	*				*

Với mỗi câu a,b: Sinh viên làm đúng (điền chính xác các ô và đánh dấu lỗi trang – nếu có) từ 5-9 cột được 0.25đ; từ 10-14 cột được 0.5đ; từ 15-19 cột được 0.75đ; 20/20 cột: 1đ.

## ĐỀ 2

Phần Trắc nghiệm (mỗi câu đúng được 0.3đ)

1C    2B    3A    4C    5D    6C    7C    8B    9C    10A  
11B    12C    13B    14D    15C    16C    17B    18A    19B    20A

Phần Tự luận:

Câu 1 (1đ):

Khai báo và khởi tạo các semaphore: (Trừ 0.25 đ nếu không khai báo và khởi tạo các semaphore sử dụng)

init(sem1,0); //khởi tạo semaphore sem1 có giá trị bằng 0

init(sem2,0); //khởi tạo semaphore sem2 có giá trị bằng 0

<pre>void T1(void) {  //T1 thực thi  signal(sem1) signal(sem1) }</pre>	<pre>void T2(void) {  wait(sem1)  //T2 thực thi  signal(sem2)  }</pre>	<pre>void T3(void) {  wait(sem1)  //T3 thực thi  signal(sem2)  }</pre>	<pre>void T4(void) {  wait(sem2) wait(sem2)  //T4 thực thi  }</pre>
------------------------------------------------------------------------	------------------------------------------------------------------------	------------------------------------------------------------------------	---------------------------------------------------------------------

Đặt đúng vị trí các hàm wait() và signal() vào các hàm thực thi bên trên được 0.25 đ/hàm thực thi.

Xét trên toàn bộ 4 hàm thực thi, nếu sinh viên đặt đúng ít hơn 3 vị trí thì được 0.25đ khuyến khích, từ 3-5 vị trí thì được 0.5đ, từ 6-7 vị trí thì được 0.75đ.

Nếu sinh viên sử dụng nhiều hơn 2 semaphore nhưng đáp ứng được yêu cầu đề bài thì vẫn được trọn điểm.

Câu 2 (1đ):

Request P4 (2, 2, 1, 3)  $\leq$  Need P4 (2, 3, 1, 5)

Request P4 (2, 2, 1, 3)  $\leq$  Available (3, 4, 4, 3)

(Trừ 0.25đ nếu sinh viên không xét hai điều kiện này)

Giả sử hệ thống đáp ứng yêu cầu thêm tài nguyên (2, 2, 1, 3) của tiến trình P4.

Trạng thái của hệ thống sau khi đáp ứng:

	Allocation				Max				Need				Available (Work)				
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	
P1	1	2	2	3	2	3	4	3	1	1	2	0	1	2	3	0	P1
P2	3	1	3	1	3	8	6	1	0	7	3	0	2	4	5	3	P4
P3	2	1	2	5	7	7	5	7	5	6	3	2	7	7	11	8	P5
P4	5	3	6	5	5	4	6	7	0	1	0	2	8	11	15	10	P2
P5	1	4	4	2	1	6	7	3	0	2	3	1	11	12	18	11	P3

Sau khi cấp phát cho P4 thì hệ thống vẫn có chuỗi an toàn <P1, P4, P5, P2, P3> cho nên hệ thống đáp ứng yêu cầu cấp phát cho P4.

- Tính đúng ma trận Need được 0.5 đ. Tính sai Need thì được 0.25đ và không chấm tiếp.

- Tính sai ma trận Available thì được 0.25đ.

- Tính đúng ma trận Available, xác định đúng chuỗi an toàn và kết luận đúng (cấp phát được) thì được 0.5đ.

- Tính đúng ma trận Available, nhưng xác định sai chuỗi an toàn hoặc kết luận sai (không cấp phát được) hoặc không kết luận gì thì được 0.25đ.

Câu 3 (2đ):

a. Giải thuật OPT có số lỗi trang là: 9

3	2	4	6	1	2	3	4	5	1	6	2	5	5	4	1	6	5	4	3
3	3	3	3	3	3	3	3	5	5	5	5	5	5	5	5	5	5	5	3
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1
		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
			6	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	6
*	*	*	*	*				*		*					*				*

b. Giải thuật LRU có số lỗi trang là: 15

3	2	4	6	1	2	3	4	5	1	6	2	5	5	4	1	6	5	4	3
3	3	3	3	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5
	2	2	2	2	2	2	2	2	1	1	1	1	1	4	4	4	4	4	4
		4	4	4	4	3	3	3	3	6	6	6	6	6	1	1	1	1	3
			6	6	6	6	4	4	4	4	2	2	2	2	2	6	6	6	6
*	*	*	*	*		*	*	*	*	*	*			*	*	*			*

Với mỗi câu a,b: Sinh viên làm đúng (điền chính xác các ô và đánh dấu lỗi trang – nếu có) từ 5-9 cột được 0.25đ; từ 10-14 cột được 0.5đ; từ 15-19 cột được 0.75đ; 20/20 cột: 1đ.

### ĐỀ 3

Phần Trắc nghiệm (mỗi câu đúng được 0.3đ)

1A    2A    3C    4B    5C    6C    7B    8A    9C    10B  
11B   12B   13C   14A   15B   16C   17A   18A   19A   20B

Phần Tự luận:

Câu 1 (1đ):

Khai báo và khởi tạo các semaphore: (Trừ 0.25 đ nếu không khai báo và khởi tạo các semaphore sử dụng)

init(sem1,0); //khởi tạo semaphore sem1 có giá trị bằng 0

init(sem2,0); //khởi tạo semaphore sem2 có giá trị bằng 0

<pre>void T1(void) {  //T1 thực thi  signal(sem1) signal(sem1) }</pre>	<pre>void T2(void) {  wait(sem1)  //T2 thực thi  signal(sem2)  }</pre>	<pre>void T3(void) {  wait(sem1)  //T3 thực thi  signal(sem2)  }</pre>	<pre>void T4(void) {  wait(sem2) wait(sem2)  //T4 thực thi  }</pre>
------------------------------------------------------------------------	------------------------------------------------------------------------	------------------------------------------------------------------------	---------------------------------------------------------------------

Đặt đúng vị trí các hàm wait() và signal() vào các hàm thực thi bên trên được 0.25 đ/hàm thực thi.

Xét trên toàn bộ 4 hàm thực thi, nếu sinh viên đặt đúng ít hơn 3 vị trí thì được 0.25đ khuyến khích, từ 3-5 vị trí thì được 0.5đ, từ 6-7 vị trí thì được 0.75đ.

Nếu sinh viên sử dụng nhiều hơn 2 semaphore nhưng đáp ứng được yêu cầu đề bài thì vẫn được trọn điểm.

Câu 2 (1đ):

Request P4 (2, 3, 1, 3)  $\leq$  Need P4 (2, 3, 1, 5)

Request P4 (2, 3, 1, 3)  $\leq$  Available (3, 4, 4, 3)

(Trừ 0.25đ nếu sinh viên không xét hai điều kiện này)

Giả sử hệ thống đáp ứng yêu cầu thêm tài nguyên (2, 3, 1, 3) của tiến trình P4.

Trạng thái của hệ thống sau khi đáp ứng:

	Allocation				Max				Need				Available (Work)				
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	
P1	1	2	2	3	2	3	4	3	1	1	2	0	1	1	3	0	P1
P2	3	1	3	1	3	8	6	1	0	7	3	0	2	3	5	3	P4
P3	2	1	4	5	7	7	5	7	5	6	1	2	7	7	11	8	P5
P4	5	4	6	5	5	4	6	7	0	0	0	2	8	11	15	10	P2
P5	1	4	4	2	1	6	7	3	0	2	3	1	11	12	18	11	P3

Sau khi cấp phát cho P4 thì hệ thống vẫn có chuỗi an toàn <P1, P4, P5, P2, P3> cho nên hệ thống đáp ứng yêu cầu cấp phát cho P4.

- Tính đúng ma trận Need được 0.5 đ. Tính sai Need thì được 0.25đ và không chấm tiếp.
- Tính sai ma trận Available thì được 0.25đ.
- Tính đúng ma trận Available, xác định đúng chuỗi an toàn và kết luận đúng (cấp phát được) thì được 0.5đ.
- Tính đúng ma trận Available, nhưng xác định sai chuỗi an toàn hoặc kết luận sai (không cấp phát được) hoặc không kết luận gì thì được 0.25đ.

Câu 3 (2đ):

a. Giải thuật OPT có số lỗi trang là: 9

3	2	4	6	1	2	3	5	6	1	2	3	5	6	4	1	6	5	4	3
3	3	3	3	3	3	3	5	5	5	5	5	5	5	5	5	5	5	5	3
	2	2	2	2	2	2	2	2	2	2	3	3	3	4	4	4	4	4	4
		4	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
			6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
*	*	*	*	*			*				*			*					*

b. Giải thuật LRU có số lỗi trang là: 16

3	2	4	6	1	2	3	5	6	1	2	3	5	6	4	1	6	5	4	3
3	3	3	3	1	1	1	1	6	6	6	6	5	5	5	5	5	5	5	5
	2	2	2	2	2	2	2	2	1	1	1	1	6	6	6	6	6	6	6
		4	4	4	4	3	3	3	3	2	2	2	2	4	4	4	4	4	4
			6	6	6	6	5	5	5	5	3	3	3	3	1	1	1	1	3
*	*	*	*	*		*	*	*	*	*	*	*	*	*	*				*

Với mỗi câu a,b: Sinh viên làm đúng (điền chính xác các ô và đánh dấu lỗi trang – nếu có) từ 5-9 cột được 0.25đ; từ 10-14 cột được 0.5đ; từ 15-19 cột được 0.75đ; 20/20 cột: 1đ.

## ĐỀ 4

Phần Trắc nghiệm (mỗi câu đúng được 0.3đ)

1B    2A    3B    4D    5D    6D    7C    8A    9C/D    10B

11B    12A    13C    14B    15D    16B    17B    18B    19C    20C

(Câu 9: Sinh viên chọn C hoặc D hoặc cả C và D đều được trọn điểm)

Phần Tự luận:

Câu 1 (1đ):

Khai báo và khởi tạo các semaphore: (Trừ 0.25 đ nếu không khai báo và khởi tạo các semaphore sử dụng)

init(sem1,0); //khởi tạo semaphore sem1 có giá trị bằng 0

init(sem2,0); //khởi tạo semaphore sem2 có giá trị bằng 0

<pre>void T1(void) {  //T1 thực thi  signal(sem1) signal(sem1) }</pre>	<pre>void T2(void) {  wait(sem1)  //T2 thực thi  signal(sem2) }</pre>	<pre>void T3(void) {  wait(sem1)  //T3 thực thi  signal(sem2) }</pre>	<pre>void T4(void) {  wait(sem2) wait(sem2)  //T4 thực thi  }</pre>
------------------------------------------------------------------------	-----------------------------------------------------------------------	-----------------------------------------------------------------------	---------------------------------------------------------------------

Đặt đúng vị trí các hàm wait() và signal() vào các hàm thực thi bên trên được 0.25 đ/hàm thực thi.

Xét trên toàn bộ 4 hàm thực thi, nếu sinh viên đặt đúng ít hơn 3 vị trí thì được 0.25đ khuyến khích, từ 3-5 vị trí thì được 0.5đ, từ 6-7 vị trí thì được 0.75đ.

Nếu sinh viên sử dụng nhiều hơn 2 semaphore nhưng đáp ứng được yêu cầu đề bài thì vẫn được trọn điểm.

Câu 2 (1đ):

Request P4 (2, 2, 1, 3)  $\leq$  Need P4 (2, 3, 1, 5)

Request P4 (2, 2, 1, 3)  $\leq$  Available (3, 4, 4, 3)



(Trừ 0.25đ nếu sinh viên không xét hai điều kiện này)

Giả sử hệ thống đáp ứng yêu cầu thêm tài nguyên (2, 2, 1, 3) của tiến trình P4.

Trạng thái của hệ thống sau khi đáp ứng:

	Allocation				Max				Need				Available (Work)				
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	
P1	1	2	2	3	2	3	4	3	1	1	2	0	1	2	3	0	P1
P2	3	1	3	1	3	8	6	1	0	7	3	0	2	4	5	3	P4
P3	2	1	2	5	7	7	5	7	5	6	3	2	7	7	11	8	P5
P4	5	3	6	5	5	4	6	7	0	1	0	2	8	11	15	10	P2
P5	1	4	4	2	1	6	7	3	0	2	3	1	11	12	18	11	P3

Sau khi cấp phát cho P4 thì hệ thống vẫn có chuỗi an toàn <P1, P4, P5, P2, P3> cho nên hệ thống đáp ứng yêu cầu cấp phát cho P4.

- Tính đúng ma trận Need được 0.5 đ. Tính sai Need thì được 0.25đ và không chấm tiếp.

- Tính sai ma trận Available thì được 0.25đ.

- Tính đúng ma trận Available, xác định đúng chuỗi an toàn và kết luận đúng (cấp phát được) thì được 0.5đ.

- Tính đúng ma trận Available, nhưng xác định sai chuỗi an toàn hoặc kết luận sai (không cấp phát được) hoặc không kết luận gì thì được 0.25đ.

Câu 3 (2đ):

a. Giải thuật OPT có số lỗi trang là: 8

3	2	4	6	5	2	1	4	6	1	6	2	5	5	4	1	6	5	4	3
3	3	3	3	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	3
	2	2	2	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	5
		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
			6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
*	*	*	*	*		*						*							*

b. Giải thuật LRU có số lỗi trang là: 13

3	2	4	6	5	2	1	4	6	1	6	2	5	5	4	1	6	5	4	3
3	3	3	3	5	5	5	5	6	6	6	6	6	6	6	1	1	1	1	3
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	6	6	6	6
		4	4	4	4	1	1	1	1	1	1	1	1	4	4	4	4	4	4
			6	6	6	6	4	4	4	4	4	5	5	5	5	5	5	5	5
*	*	*	*	*		*	*	*				*		*	*	*			*

Với mỗi câu a,b: Sinh viên làm đúng (điền chính xác các ô và đánh dấu lỗi trang – nếu có) từ 5-9 cột được 0.25đ; từ 10-14 cột được 0.5đ; từ 15-19 cột được 0.75đ; 20/20 cột: 1đ.