

(Sinh viên không được sử dụng tài liệu. Làm bài trực tiếp trên đề)

HỌ VÀ TÊN SV: MSSV: STT: PHÒNG THI:	<u>ĐIỂM</u>	<u>CÁN BỘ CỎI THI</u>
--	--------------------	------------------------------

PHẦN 1. TRẮC NGHIỆM (6.0đ)

Sinh viên chọn câu trả lời chính xác nhất và điền vào bảng trả lời sau:

Câu 1	Câu 2	Câu 3	Câu 4	Câu 5
Câu 6	Câu 7	Câu 8	Câu 9	Câu 10
Câu 11	Câu 12	Câu 13	Câu 14	Câu 15

1. Công cụ nào dưới đây **KHÔNG** được dùng để chuyển dữ liệu giữa các tiến trình trên Linux? (G1)
- A. Pipe. B. Network.
- C. Shared memory. D. Signal.

2. Xét một hệ thống máy tính có 5 tiến trình: P1, P2, P3, P4, P5 và 4 loại tài nguyên: R1, R2, R3, R4. Tại thời điểm t_0 , trạng thái của hệ thống như sau:

	Allocation				Max			
Tiến trình	R1	R2	R3	R4	R1	R2	R3	R4
P1	1	1	2	2	3	2	3	3
P2	3	1	4	2	3	5	6	4
P3	5	2	1	2	7	5	4	5
P4	3	2	3	3	3	4	5	6
P5	1	2	4	1	3	6	4	7

Available			
R1	R2	R3	R4
3	4	3	3

- Lựa chọn nào dưới đây **KHÔNG** phải là một chuỗi an toàn của hệ thống? (G1)
- A. <P4, P5, P2, P3, P1>
B. <P2, P3, P1, P4, P5>
C. <P3, P1, P2, P5, P4>
D. <P1, P5, P4, P2, P3>

3. Chọn phát biểu **SAI** về semaphore? (G1)
- A. Nếu giá trị của semaphore S bằng -4 thì có 4 tiến trình đang bị blocked trong hàng đợi của S.
 - B. Thao tác signal() sẽ được sử dụng khi muốn sử dụng tài nguyên.
 - C. Tiến trình sẽ phải chờ khi thực hiện thao tác wait() lên semaphore đang có giá trị không dương.
 - D. Semaphore được phân loại thành counting semaphore và binary semaphore.

4. Lựa chọn nào dưới đây **KHÔNG** phải là điều kiện cần để xảy ra deadlock? (G1)
- A. Một tiến trình đang giữ ít nhất một tài nguyên và đợi thêm tài nguyên do tiến trình khác giữ.
 - B. Tài nguyên không thể bị lấy lại mà chỉ có thể được trả lại từ tiến trình đang giữ tài nguyên đó khi nó muốn.
 - C. Ít nhất một tài nguyên được giữ theo chế độ không chia sẻ.
 - D. Mỗi tiến trình chỉ có thể yêu cầu thực thể của một loại tài nguyên theo thứ tự tăng dần của loại tài nguyên.

5. Giải pháp đồng bộ nào sau đây không cần sự hỗ trợ từ hệ điều hành? (G1)
- A. Semaphore.
 - B. Monitor.
 - C. Mutex locks.
 - D. Giải thuật Peterson.

Sử dụng các dữ liệu sau để trả lời câu hỏi 6, 7:

Giả sử một tiến trình được cấp 5 khung trang trong bộ nhớ vật lý và 7 trang trong bộ nhớ ảo. Tại thời điểm nạp tiến trình vào, 5 khung trang trên bộ nhớ vật lý này đang trống. Tiến trình truy xuất 7 trang (1, 2, 3, 4, 5, 6, 7) trong bộ nhớ ảo theo thứ tự như sau:

1 2 1 5 4 3 7 2 6 5 3 4

6. Tại thời điểm tiến trình truy xuất trang nhớ số 7 lần đầu tiên, trang nhớ nào sẽ được chọn làm trang hy sinh (victim page), nếu sử dụng giải thuật thay thế trang LRU? (G1)
- A. 1
 - B. 2
 - C. 5
 - D. 4
7. Có tất cả bao nhiêu lỗi trang xảy ra nếu tiến trình sử dụng giải thuật thay thế trang OPT? (G1)
- A. 4
 - B. 5
 - C. 6
 - D. 7
8. Xét bài toán Reader-Writer có sử dụng 2 semaphore (rw_mutex và mutex) để đồng bộ. Các semaphore được khởi tạo bằng 1. Cho mã nguồn của các tiến trình Reader, Writer như bên dưới. Biết rằng giải pháp này ưu tiên Reader và readcount là biến chia sẻ được khởi tạo bằng 0.

Reader

```
wait(mutex);
read_count++;
if (read_count == 1) /* first reader */
    wait(rw_mutex);
signal(mutex);
...
/* đọc dữ liệu */
...
wait(mutex);
read_count--;
if (read_count == 0) /* last reader */
    signal(rw_mutex);
signal(mutex);
```

Writer

```
wait(rw_mutex);
...
/* cập nhật dữ liệu */
...
signal(rw_mutex);
```

Chọn phát biểu **SAI** trong các phát biểu dưới đây? (G1)

- A. Nếu một Writer đang ở trong vùng tranh chấp và có n Readers đang đợi thì một Reader được xếp trong hàng đợi của rw_mutex và n – 1 Reader kia trong hàng đợi của mutex.
- B. Khi Writer thực thi signal(rw_mutex), hệ thống có thể phục hồi thực thi của một trong các Reader đang đợi hoặc Writer đang đợi.
- C. Các lệnh readcount++ và readcount-- đều là các vùng tranh chấp.
- D. Semaphore mutex được sử dụng để bảo đảm tính chất loại trừ tương hỗ đối với các Writer.

9. Chiến lược next-fit sẽ chọn khối nhớ trống như thế nào? (G1)
- A. Chọn khối nhớ trống nhỏ nhất.
 - B. Chọn khối nhớ trống phù hợp đầu tiên kể từ đầu bộ nhớ.
 - C. Chọn khối nhớ trống phù hợp đầu tiên kể từ vị trí cấp phát cuối cùng.
 - D. Chọn khối nhớ trống lớn nhất.

10. Hệ thống nào trong các hệ thống bên dưới có deadlock xảy ra? (G1)
- (1) Hệ thống có 3 tiến trình P1, P2, P3 và 2 loại tài nguyên R1, R2 – mỗi loại có 1 thực thể. P1 giữ 1 R1 và yêu cầu 1 R2, P2 giữ 1 R2, P3 yêu cầu 1 R2.
- (2) Hệ thống có 3 tiến trình P1, P2, P3 và 3 loại tài nguyên R1, R2, R3 – mỗi loại có 1 thực thể. P1 giữ 1 R1 và yêu cầu 1 R3, P2 giữ 1 R2 và yêu cầu 1 R3, P3 giữ 1 R3 và yêu cầu 1 R1.
- (3) Hệ thống có 3 tiến trình P1, P2, P3 và 2 loại tài nguyên R1, R2 – mỗi loại có 2 thực thể. P1 yêu cầu 1 R1 và 1 R2, P2 giữ 1 R1 và 1 R2, P3 giữ 1 R1 và yêu cầu 1 R2.
- A. (1) B. (1) và (2) C. (2) D. (3)

11. Chọn phát biểu **SAI** về việc cấp phát khung trang cho tiến trình? (G1)
- A. Nếu hệ điều hành cấp ít khung trang cho tiến trình thì sẽ làm xuất hiện nhiều lỗi trang.
- B. Nếu hệ điều hành cấp nhiều khung trang cho tiến trình thì sẽ làm giảm mức độ đa chương.
- C. Số lượng khung trang được cấp cho mỗi tiến trình sẽ không thay đổi nếu sử dụng chiến lược cấp phát tĩnh.
- D. Khi sử dụng chiến lược cấp phát động, nếu tỉ lệ lỗi trang cao, hệ điều hành sẽ giảm bớt số lượng khung trang được cấp.

12. Lựa chọn nào dưới đây **KHÔNG** phải là một yêu cầu của bài toán bounded-buffer? (G1)

- A. Producer và Consumer không được thao tác trên buffer cùng lúc.
- B. Producer không được ghi dữ liệu vào buffer đã đầy
- C. Consumer không được đọc dữ liệu từ buffer đang trống.
- D. Luôn có ít nhất một Producer hoặc Consumer thao tác trên buffer tại bất cứ thời điểm nào.

Sử dụng các dữ liệu sau để trả lời câu hỏi 13, 14:

Xét một không gian địa chỉ ảo có 256 trang, mỗi trang có kích thước 4096 byte được ánh xạ vào bộ nhớ vật lý có 64 khung trang.

13. Địa chỉ vật lý gồm bao nhiêu bit? (G1)

- A. 18 B. 12 C. 20 D. 8

14. Kích thước bảng phân trang là bao nhiêu byte, nếu mỗi mục (entry) trong bảng phân trang cần 2 byte? (G1)

- A. 128 B. 4096 C. 512 D. 32

15. Memory barrier là gì? (G1)

- A. Mô hình hoạt động của bộ nhớ trong hệ thống, bao gồm cách thức quản lý và truy xuất đến các vùng nhớ được cấp phát cho các tiến trình và luồng trong hệ thống.
- B. Một chỉ thị (instruction) mà bắt buộc mọi thay đổi trong bộ nhớ phải được truyền tải (hiển thị) đến tất cả bộ xử lý khác.
- C. Tập hợp các quy tắc và ràng buộc cho việc sử dụng bộ nhớ.
- D. Cơ chế cho phép vi xử lý hoặc trình biên dịch sắp xếp lại các thao tác độc lập với nhau trên bộ nhớ.

PHẦN 2. TỰ LUẬN – (G1)

Câu 1 (1đ): Xác định các phát biểu dưới đây là ĐÚNG hay SAI? (Lưu ý: Chỉ cần trả lời là **Đúng** hoặc **Sai**, không cần giải thích gì thêm).

- a. Hiện tượng trì trệ (thrashing) xuất hiện khi tổng kích thước các cục bộ (locality) lớn hơn kích thước bộ nhớ vật lý.

Trả lời:

- b. Giải thuật ngăn deadlock sẽ kiểm tra trạng thái cấp phát tài nguyên để đảm bảo hệ thống không rơi vào deadlock.

Trả lời:

- c. Cơ chế kết khối (compaction) có thể được sử dụng để giải quyết vấn đề phân mảnh nội.

Trả lời:

Trả lời:

a. Kỹ thuật nào cho phép xử lý một tiến trình không được nạp toàn bộ vào bộ nhớ vật lý?

Trả lời:

Trả lời:

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Xét một hệ thống có 2 tiểu trình T1, T2 với các hàm thực thi được minh họa như bên dưới.

```
void* T1(void* arg)
{
    printf("1");
    printf("2");
    printf("5");
    printf("6");
}
```

```
void* T2(void* arg)
{
    printf("3");
    printf("4");
    printf("7");
    printf("8\n");
}
```

Hãy sử dụng semaphore để đồng bộ hoạt động của các tiểu trình này để đảm bảo khi thực thi chúng thì kết quả sau sẽ được in ra:

12345678

Khai báo và khởi tạo giá trị cho các semaphore cần sử dụng:

.....

.....

.....

.....

Đặt các lệnh wait() và signal() tương ứng với semaphore cần dùng vào hàm thực thi của mỗi tiểu trình để đảm bảo yêu cầu đồng bộ trên.

(Lưu ý:

- Chỉ điền vào các dòng trống được ký hiệu từ [A] đến [F].
- Có thể điền nhiều lệnh tại mỗi dòng.
- Có thể để trống nếu thấy không cần phải điền.
- Nếu không đủ chỗ trống để điền thì có thể viết sang trang 6.)

```
void* T1(void* arg)
{
[A].....

.....

    printf("1");
    printf("2");
[B].....

.....

    printf("5");
    printf("6");
[C].....

.....

}
```

```
void* T2(void* arg)
{
[D].....

.....

    printf("3");
    printf("4");
[E].....

.....

    printf("7");
    printf("8\n");
[F].....

.....

}
```

