دانشکده مهندسی کامپیوتر

درس سیستم‌های عامل

پاسخنامه کوییز چهارم

مدرس ..... دکتر رضا انتظاری ملکی

طراح ..... محمدحسین عباسپور

۱) با توجه به deadlock prevention و deadlock avoidance کدام گزینه درست نمی‌باشد؟

۱. در deadlock prevention، اگر حالت بعدی safe state باشد، آنگاه تخصیص منابع همواره اعطا خواهد شد.
۲. در deadlock avoidance، اگر حالت بعدی safe state باشد، آنگاه تخصیص منابع همواره اعطا خواهد شد.
۳. در deadlock avoidance یک سری اطلاعات قبلی مانند بیشینه منابع مورد نیاز برای جلوگیری از deadlock نیاز می‌باشد.
۴. در deadlock prevention با نقض شرط Hold and Wait احتمال گرسنگی فرآیندها (starvation) وجود دارد.

۲) کدام گزینه درست می‌باشد؟

۱. در deadlock prevention و deadlock avoidance سیستم وارد حالت deadlock می‌شود و سپس با استفاده از یک سری پروتکل، سیستم را بازیابی و از حالت deadlock خارج می‌کنیم.
۲. یک پروتکلی که برای باطل کردن شرط No Preemption در روش deadlock prevention استفاده می‌شود این است که در صورتی که یک فرآیند به یک منبع جدید نیاز داشت و آن منبع در دسترس بود، منابع قبلی را نگه داشته و منبع جدید را به دست می‌آورد.
۳. هم در safe state و هم در unsafe state احتمال رخ دادن deadlock وجود دارد.
۴. در یک سیستم با یک منبع واحد، در صورتی deadlock رخ می‌دهد که بیشتر از ۱ فرآیند برای آن منبع با هم رقابت کنند.

۳) یک سیستم را با ۸ منبع در نظر بگیرید که n فرآیند برای آنها با هم رقابت می‌کنند. هر فرآیند به ۳ منبع نیاز دارد. کدام یک از گزینه‌های زیر صحیح می‌باشد؟ (ممکن است بیشتر از یک گزینه صحیح باشد)

۱. بیشترین مقدار n برای اینکه سیستم هیچگاه وارد deadlock نشود برابر ۳ است.
۲. اگر n برابر ۴ باشد، آنگاه احتمال رخ دادن deadlock برای سیستم وجود دارد.
۳. بیشترین مقدار n برای اینکه سیستم هیچگاه وارد deadlock نشود برابر ۲ است.
۴. اگر n برابر ۳ باشد، آنگاه احتمال رخ دادن deadlock برای سیستم وجود دارد.

۴) فرآیندهای P_1، P_2، P_3 ، ... وP_nرا در یک سیستم با R منبع در نظر بگیرید. فرآیند P_1 به اندازه x_1، فرآیند P_2 به اندازه x_2، فرآیند P_3 به اندازه x_3، .... و فرآیند P_n به اندازه x_n به منبع نیاز دارند. کمترین مقدار R چند باشد تا احتمال رخ دادن deadlock در سیستم صفر باشد؟

۱. $\sum_{i=1}^{i=n} x_i$
۲. $\left(\sum_{i=1}^{i=n} x_i\right) - n$
۳. $\left(\sum_{i=1}^{i=n} x_i\right) - n + 1$
۴. $\left(\sum_{i=1}^{i=n} x_i\right) + 1$

۵) یک سیستم دارای ۴ منبع A, B, C, D می‌باشد. تعداد واحدهای اولیه از هر منبع به شکل زیر می‌باشد:

A = 5, B = 4, C = 4, D = 4

اسنپ‌شات زیر را از سیستم در نظر بگیرید:

	Allocation A B C D	Request A B C D
P0	2 0 1 3	0 2 0 0
P1	0 2 1 1	1 1 2 1
P2	1 1 1 0	1 1 0 0

با استفاده از الگوریتم deadlock-detection مشخص کنید که آیا سیستم دچار deadlock می‌شود؟ اگر آری کدام فرآیندها درگیر deadlock می‌شوند و اگر نه دنباله امن را پیدا کنید.

۶) اسنپ‌شات زیر را برای سیستم در نظر بگیرید:

	Allocation A B C D	Max A B C D
T0	3 0 1 4	5 1 1 7
T1	2 2 1 0	3 2 1 1
T2	3 1 2 1	3 3 2 1
T3	0 5 1 0	4 6 1 2
T4	4 2 1 2	6 3 2 5

با استفاده از الگوریتم بانکدار تعیین کنید که آیا حالت زیر یک حالت امن می‌باشد یا ناامن. اگر در حالت امن بود دنباله امن را بنویسید و اگر در حالت ناامن بود توضیح دهید که چرا ناامن است.

Available = (1, 0, 0, 2)

پاسخنامه

(۱

۱. غلط. در deadlock prevention به دنبال باطل کردن یکی از شروط deadlock هستیم و بحث حالت امن مطرح نیست و اگر حالت بعدی امن باشد نمی‌توان در مورد همواره اعطا کردن منبع تصمیم گرفت.
۲. صحیح. در deadlock avoidance با استفاده از روش‌هایی مانند الگوریتم بانکدار به دنبال این هستیم که سیستم از حالت امن خارج نشود. پس اگر تخصیص منبعی باعث شود که حالت بعدی امن باشد آن منبع همواره اعطا خواهد شد.
۳. صحیح. برای مشخص کردن امن با ناامن بودن حالت بعدی نیاز به یک سری اطلاعات مانند حداکثر منبع مورد نیاز فرآیندها داریم.
۴. صحیح. چون در یکی از پروتکل‌های باطل کردن این شرط داریم که اگر فرآیندی نیاز به یک منبع جدید داشت آنگاه تمام منابع در دسترسش را آزاد می‌کند تا دوباره تمام این منابع بعلاوه منبع جدید را به دست آورد که ممکن است باعث starvation شود. (اسلاید deadlock صفحه ۱۳)

(۲

۱. غلط .در این دو روش هیچگاه اجازه نمی‌دهیم که سیستم دچار deadlock شود.(اسلاید deadlock صفحه ۱۲)
۲. صحیح. یکی از روش‌ها برای باطل کردن شرط Hold and Wait این بود که اگر فرآیندی نیاز به یک منبع جدید داشت، چه آن منبع در دسترس باشد و چه نباشد، آن فرآیند تمام منابعش را آزاد می‌کند و دوباره تمام منابع را از اول می‌گیرد. برای نقض شرط No Preemption روشی مشابه همین می‌باشد با این تفاوت که اگر منبع درخواستی جدید در دسترس بود بدون آزاد کردن منابع قبلی، فرآیند، منبع جدید را می‌گیرد و اگر در دسترس نبود آنگاه تمام منابعش را آزاد می‌کند. (اسلاید deadlock صفحه ۱۴)
۳. غلط. در حالت امن deadlock رخ نمی‌دهد. (اسلاید deadlock صفحه ۱۷)
۴. غلط. deadlock زمانی رخ می‌دهد که هر ۴ شرط mutual exclusion و hold and wait و no preemption و circular wait با هم برقرار باشند. هنگامی که فقط یک منبع داریم، شروط hold and wait و circular wait نقض می‌شوند. با فرض اینکه هیچ فرآیندی یک منبع را برای زمان بی‌نهایت در اختیار

نمی‌گیرد، بالاخره یک فرآیند تمام می‌شود و فرآیندهای دیگر میتوانند منبع را بگیرند. در نتیجه deadlock هیچ وقت رخ نخواهد داد.

۳)

اگر یک فرآیند نیاز به k منبع داشته باشد؛ آنگاه بیشترین مقدار منبع مورد نیاز که ممکن است باعث deadlock شود برابر k-1 می‌باشد.

اگر تعداد فرآیندها برابر ۳ باشد:

Processes	Required resources	Max number of resources to cause deadlock
P1	3	2
P2	3	2
P3	3	2

۶=۲+۲+۲ و ۲ منبع آزاد می‌ماند پس deadlock رخ نمی‌دهد.

اگر تعداد فرآیندها برابر ۴ باشد:

Processes	Required resources	Max number of resources to cause deadlock
P1	3	2
P2	3	2
P3	3	2
P4	3	2

۸=۲+۲+۲+۲ و هیچ منبعی آزاد نمی‌ماند. پس deadlock ممکن است رخ دهد. یکی از حالات که deadlock رخ می‌دهد این است که به تمام فرآیندها ۲ منبع بدهیم در نتیجه هیچ منبع آزادی باقی نمی‌ماند و هر فرآیندی برای اتمام نیاز به یک منبع دارد که در اختیار دیگر فرآیندها است.

در نتیجه گزینه‌های ۱ و ۲ صحیح می‌باشند.

۴)

بیشترین مقدار منبع که باعث deadlock برای فرآیند P_i میشود برابر $x_i - 1$ می‌باشد. در نتیجه بیشترین مقدار منبع که ممکن است باعث deadlock برای سیستم شود برابر است با:

$$(x_1 - 1) + (x_2 - 1) + (x_3 - 1) + \ldots (x_n - 1) = (x_1 + x_2 + x_3 + \ldots + x_n) - n$$
$$= \sum_{i=1}^{i=n} x_i \; - \; n$$

کمترین مقدار منبع مورد نیاز برای جلوگیری از deadlock برابر است با بیشترین منبع مورد نیاز برای وجود احتمال deadlock بعلاوه ۱. (یک منبع بیشتر باعث می‌شود که یکی از فرآیندها با گرفتن آن به اتمام برسد و منابعش را آزاد کند در نتیجه بقیه فرآیندها هم به اتمام می‌رسند)

$$R = \left(\sum_{i=1}^{i=n} x_i \right) \; - \; n \; + \; 1$$

(۵

①	Allocation A B C D	Request A B C D	Available
P_0	2 0 1 3	0 2 0 0	2, 1, 1, 0
P_1	0 2 1 1	1 1 2 1	
✓ P_2	1 1 1 0	1 1 0 0 → select P_2	

②	Allocation A B C D	Request A B C D	Available
✓ P_0	2 0 1 3	0 2 0 0 → select P_0	3, 2, 2, 0
P_1	0 2 1 1	1 1 2 1	

③	Allocation A B C D	Request A B C D	Available
✓ P_1	0 2 1 1	1 1 2 1	5, 2, 3, 3

safe state → sequence = $\langle P_2, P_0, P_1 \rangle$

۶)

	Allocation	Max	Available	Need
③ T_0	3 0 1 4	5 1 1 7	1 0 0 2	2 1 0 3
① T_1	2 2 1 0	3 2 1 1		1 0 0 1
② T_2	3 1 2 1	3 3 2 1		0 2 0 0
④ T_3	0 5 1 0	4 6 1 2		4 1 0 2
⑤ T_4	4 2 1 2	6 3 2 5		2 1 1 3

$T_1 \rightarrow A[3,2,1,2]$, $T_2 \rightarrow A[6,3,3,3]$, $T_0 \rightarrow A[9,3,4,7]$

$T_3 \rightarrow A[9,8,5,7]$, $T_4 \rightarrow A[13,10,6,9]$ → safe sequence