

به سوالات زیر پاسخ کامل دهید.

(آ) در الگوریتم ذکر (تلاش سوم) با استفاده از چه تغییراتی نسبت به تلاش دوم ویژگی انحصار متقابل رعایت شد؟  
در تلاش دوم داشتیم که هر فرآیند ابتدا شرایط فرآیند دیگر را چک می‌کند و بعد flag خودش را TRUE می‌کند. ایده این است: فرآیند ابتدا بررسی میکند که آیا اجازه‌ی ورود به حلقه برای اجرای «ناحیه بحران» را دارد یا خیر. اگر این اجازه را داشت به فرآیند دیگری اجازه‌ی ورود به ناحیه بحرانی را نمی‌دهد.

مشکل این روش آنجاست که در زمانبندی قبضه‌ای هر لحظه امکان قبضه شده یک فرآیند و دادن پردازنده به فرآیند دیگری وجود دارد. بنابراین منطقی است که هر فرآیند ابتدا مطمئن شود که فرآیند دیگر در ناحیه‌ی بحرانی خودش به هیچ وجه نخواهد رفت و سپس برای ورود به ناحیه‌ی بحرانی خودش اقدام کند. ایده‌ی روش سوم از همین زاویه دید نشأت گرفته است. یعنی: یک فرآیند، در وهله اول به فرآیند های دیگر اجازه‌ی ورود نمی‌دهد و سپس اجازه‌ی ورود خودش را بررسی می‌کند. این تغییر به سادگی و فقط با جابه‌جا کردن دو خط از روش دوم میسر می‌شود.

چرا دیگر مشکل قبضه کردن را نخواهیم داشت؟ چون ابتدا فرآیند دیگر منع می‌شود از ورود به ناحیه‌ی بحرانی.

(ب) تفاوت گرسنگی با بن بست را توضیح دهید.

**بن بست (Deadlock):** شرایطی است که چند فرآیند در رقابت برای دسترسی به منابع موردنیازشان تا ابد اجازه‌ی پیش‌روی نداشته باشند. در این شرایط تعامل بین چندفرآیند (نه فقط یکی) است که بن بست را رقم زده است. در ضمن این عدم دسترسی به منبع موردنیاز تا ابد ادامه پیدا می‌کند اگر تمهیدی برای آن در نظر گرفته نشده باشد.

**گرسنگی (Starvation):** وقتی یک فرآیند تحت شرایطی (احتمالاتی/قطععی) امکان دسترسی به یک منبع را تا زمانی مشخص به دست نیاورد.

تفاوت کلیدی بین مفهوم گرسنگی و بن بست در اینست که گرسنگی حالتی محدودتر و با قیود بیشتر از گرسنگی است. در بن بست چندفرآیند تا ابد به مشکل دسترسی به منابع برمیخورند.

وقوع یک بن بست به معنی وقوع یک گرسنگی است. چون حداقل یک فرآیند داریم که تا ابد به منبع موردنیازش دسترسی پیدا نخواهد کرد. برعکس این گزاره لزوماً برقرار نیست. یعنی ممکن است گرسنگی رخ بدهد و بن بست رخ نداده باشد.

(ج) دستورالعمل TSL و دستورالعمل Swap را مقایسه کنید.

**دستور TSL:** یک عملیات اتمیک (تقسیم ناپذیر) است. به این معنی که یا هر دو دستور مذکور اجرا میشوند یا هیچکدام اجرا نمیشوند (یا همه یا هیچ). ابتدا محتویات یک خانه از حافظه را خوانده و در یک ثبات ذخیره میکند. سپس مقدار یک را در دقیقاً همان آدرس از حافظه ذخیره میکند.

tsl reg, lock

**دستور swap:** این دستور فقط از یک عمل تقسیم ناپذیر تشکیل شده است. محتویات یک رجیستر را با محتویات یک خانه از حافظه‌ی اصلی جابه‌جا می‌کند.

swap reg, lock

نکته: دستورات ذکر شده خود دارای یک سری عملیات‌های کوچک‌تر به صورت ضمنی هستند و اجرای این مجموعه دستورات ضمنی است که اتمیک است.

(د) زمان بندی SRT چه ویژگی هایی دارد و تفاوت آن با SJF چیست؟  
این دو زمانبندی بسیار به هم شباهت دارند. ایده‌ی اصلی در قبضه‌ای یا غیرقبضه‌ای بودن آنها است. در SJF زمان‌بند، فرآیند با کوتاه‌تری زمان انفجار را انتخاب می‌کند و سپس پردازنده را به آن اختصاص می‌دهد. این زمان بندی از نوع غیر قبضه‌ای است. یعنی فرآیند تا زمان خاتمه یافتن به اجرای خودش ادامه می‌دهد. روش پیاده سازی این روش با درخت هرم کمینه (min-heap tree) به سادگی میسر می‌شود.  
از طرفی SRT همان روش قبلی با پیاده‌سازی پیچیده‌تر و عملکردی منعطف‌تر است. این مقایسه‌ی زمان انفجار پیوسته در حال انجام است. دو نکته را باید در پیاده سازی مد نظر قرار داد و آن را مدیریت کرد:

- فرآیند در حال اجرا که به صورت پیوسته از زمان انفجارش کم می‌شود.
  - یک فرآیند جدید وارد صف آماده شود.
- برخلاف روش قبل که پردازنده از فرآیند تا اتمام آن یا وقوع وقفه حین اجرا قبضه نمی‌شد، هنگام ورود به یک فرآیند جدید، درخت هرم کمینه بروزرسانی می‌شود و در پایان این بروزرسانی فرآیند با کمترین زمان انفجار در ریشه‌ی درخت است و زمان بند آن را انتخاب کند و پردازنده را به آن اختصاص دهد. این فرآیند ممکن است که فرآیند در حال اجرا و یا فرآیند تازه وارد باشد.