۱. بن بست حالتی است که در آن همه پردازه ها منتظر اتمام کار یک دیگر میمانند.
شرایط:

۱. حداقل یک منبع نباید به صورت اشتراکی استفاده شود

۲. شرط hold and wait: در هنگام درخواست منبع برخی از پردازه ها منابعی را در اختیار داشته باشند.

۳. preemption: برخی منابع را نتوان به اجبار از پردازه ها گرفت.

۴.circular wait: چرخه ای از پردازه های منتظر به وجود بیاید که هر پردازه منتظر پردازه بعدی است.

۲. ۱. Prevention:

همه منابع به صورت اشتراکی استفاده شوند.

در هنگام درخواست منابع پردازه ها همه منابع خود را باهم در خواست دهند یعنی در شروع اجراهر پردازه منابع مورد نیازش را درخواست دهد.

برای منابعی که می توان وضعیت آن ها را ذخیره و بازیابی کرد امکان preemption

پردازه p(i) می تواند R(j)را درخواست دهد فقط اگر k>j هیچ R(j) را در اختیار نداشته باشد

deadlock avoidance:

وضعیت سیستم همیشه در حالت امن باشد. استفاده از الگوریتم بانکدار برای بررسی وضعیت امن ، به این صورت که آرایه finished را تغییر داده و در نهایت چک میکند که اگر همه true بودند در وضعیت امن است، و اینکه می توان به یک درخواست پاسخ داد یا خیر.

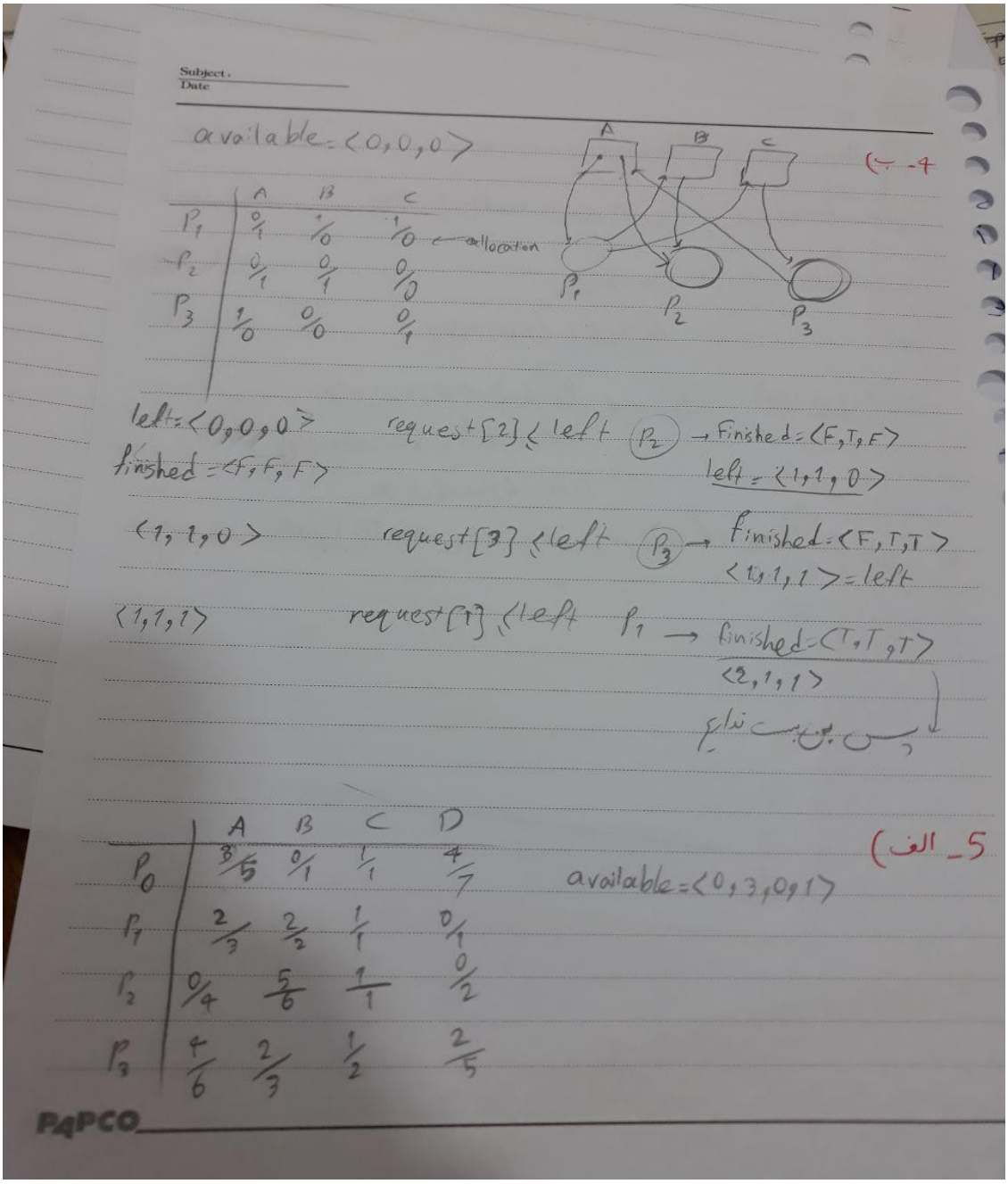
Deadlock detection: از هر منبع فقط یک نمونه. استفاده از الگوریتم مشابه بانکدار برای تشخیص بن بست و سپس حل بن بست به روش های خاتمه دادن به پردازه ها یا به اجبار گرفتن منابع. به این صورت است که با استفاده از چند ارایه در نهایت وضعیت آرایه finishedرا که برای همه پردازه ها است تغییر می دهد که اگر همه trueبودند بن بست نیست.

روش شترمرغ: نادیده گرفتن بن بست.

۳. زیرا در بدترین حالت که بین دوپردازه دور به وجود آید،باز هم میتوان یک دنباله امن بین هر ۳ پیدا کرد مثلا حالتی که بین p1, p2ود دارد، دنباله امن p1 , p2 , p3 وجود خواهد داشت. طبق الگوریتم مشاب بانکدار نیز در مرحله اول هیچ یک از پردازه ها آرایه requestکمتر از available = <0,0,0 ندارد

۴. الف)دور وجود دارد و دنباله امن نداریم پس بن بست رخ میدهد.

ب)



۵.

Date

left = <0,3,0,1> finished = <F,F,F,F>

برای هیچکدام need[i] ≤ left پس ناامن است چون finished ما همه false

(ب)

available = <1,0,0,2>

left = <1,3,0,2> finished = <F,F,F,F>

need[1] ≤ left (P_1) → finished = <F,T,F,F>

left = left + allocation[1]

left = <3,2,1,2>

برای بقیه need[i] ≤ left نمی باشد، چون هنوز finished ها false هستند پس ناامن است

۶. در چند پارگی داخلی حافظه خالی ای که وجود دارد ولی قابل استفاده نیست درون قطعه های حافظه تخصیص داده شده است.
در چند پارگی خارجی حافظه خالی موجود پیوسته نیست و در بیرون قطعه های تخصیص داده شده بین آن ها قرار دارد.

۷. الف)درست است، در برخی از الگوریتم ها رخ می دهدکه به آن ناهنجاری belady گویند
ب) درست است، بیشتر شدن اندازه صفحه باعث بیشتر شدن اندازه حافظه منطقی می شود که باعث بزرگ تر شدن اندازه جدول صفحه می شود.
ج) درست است، اگر در شروع اجرا هر پردازه منابع مورد نیازش را درخواست دهد بن بست به وجود نمی آید.
د)نادرست است، مشکل چند پارگی داخلی حل می شود.

۸ الف)چون ۶۴ صفحه ای است، پس تعداد سطرهای جدول صفحه ۶۴ است که با ۶ بیت میتوان ساخت. با توجه به اندازه صفحه که $۲^{۱۱}$ بایت است، با ۱۱ بیت میتوان offsetرا تعیین کرد. پس در آدرس منطقی ۱۷ بیت وجود دارد.
ب)offset تغییری نمیکند. با توجه به تعداد قاب ها،با ۵ بیت میتوان ۳۲ قاب داشت که درنتیجه فضای آدرس فیزیکی ۱۶ بیت است.