_____تمرین ۱_____

الف) اگر فرض کنیم که ماشین ها نمیتوانند به سمت عقب حرکت کنند در این حالت DEADLOCK رخ داده است.

ب)

- Mutual exclusion : اگر جایی که یک ماشین در خیابان میگیرد را منبع در نظر بگیریم خودرو ها در لحظه نمیتوانند در یک مکان باشند به همین دلیل همزمان نمیتوانند از یک منبع استفاده کنند به همین دلیل این شرط برقرار است.
- Hold and wait: اگر خودرویی در پشت خودرو دیگری باشد هر خودرو در منبعی است و خورو دیگر دارای منبع دیگری است به همین دلیل خودرویی که در آن نقطه است نمیتواند منبع خود را رها کند و منتظر منبع بعدی میماند به همین دلیل این شرط نیز برقرار است.
- Non-preemption: خودرو خودش باید حرکت کند و نه این که یک نیروی خارجی بتواند آن را جابه‌جا کند به همین دلیل این شرط برقرار است.
- Circular wait: همانطور که میبینیم در تصویر یک خودرو منتظر جلویی خود است و همین طور میچرخد و میبینیم در واقع منتظر حرکت خودش است.

ج)اگر شرط کنیم که تنها در حالتی وارد تقاطع شویم که خودروها در دیگر نقاط تقاطع در حال حرکت باشند. در این حالت تضمین میشود که خودرویی منتظر دیگران نباشد.

_____تمرین ۲_____

الف)

**کارایی پردازنده از زیاد به کم:**

- رزرو تمام منابع
- شماره دهی
- الگوریتم بانکدارها
- کشف بن بست و از بین بردن تمام فرایند ها
- شروع مجدد در صورت انتظار
- کشف بن بست و به عقب برگرداندن

**اجرای موازی از زیاد به کم:**

- کشف بن بست و به عقب برگرداندن
- شروع مجدد یک فرایند

- کشف بن‌بست و از بین بردن تمام فرایند ها
- الگوریتم بانکدارها
- شماره دهی
- رزرو تمام منابع از قبل

**توضیحات رده بندی بالا :**

**الگوریتم بانکداران:** در این الگوریتم باید بررسی کرد که هر وقتی میخواهیم منبعی به کسی بدهیم باید بررسی شود که ممکن است تحت این شرایط deadlock رخ دهد یا خیر؟ این بررسی در بدترین حالت در پیچیدگی زمانی $O^2$ است ضمنا این موضوع که به صورت بدبینانه ترین حالت ممکن منبع تخصیص میدهیم باعث میشود که فرایند ها کمتر بتوانند همگی با هم منبع بگیرند. و موازی سازی برنامه کمتر شود.

**کشف بن بست و از بین بردن فرایندهای دخیل:** بایستی گراف وابستگی ها رسم شود و باید به کمک طی کردن گراف بررسی شود که آیا دور تشکیل شده است یا خیر که میدانیم حرکت در گراف و ایجاد آن هزینه بالایی دارد. برای این که بخواهیم تمام فرایندهای دخیل رو از بین ببریم این کار هزینه بالایی دارد و ممکن است که فرایند های زیادی نابود شوند.ضمنا اگر بخواهیم همان فرایندها را مجددا ایجاد کنیم میبایست که از ابتدا دوباره فرایند را پردازش کنیم این موضوع دوباره سبب میشود که مقدار سربار موضوع بیشتر هم میشود. از بین بردن فرایندها باعث میشود فرایند های کاندید کمتر بشوند و این موضوع باعث میشود امکان موازی شدن فرایند ها کمتر بشود.

**رزرو تمام منابع قبل از شروع کار:** این کار سربار را کاهش میدهد ولی مشکل آن این است که میزان امکان موازی کاری را به اندازه قابل توجهی کاهش میدهد.

**شروع مجدد یک فرایند:** در این حالت هر زمان که که فرایندی منتظر باشد همه منابع خود را آزاد میکند. و بعدا کارش را مجددا شروع میکند. در این حالت سربار پردازنده زیاد است. زیرا بسیاری از کارها دوبار باید انجام شود. این امکان فراهم است که فرایندها به صورت موازی انجام شود.

**شماره دهی:** به این ترتیب فرایندها از منابع به ترتیب داده شود. و فرایندها یکی یکی منابع را در اختیار میگیرند. چون نوبتی میروند سراغ منابع پردازش موازی کاهش میابد. اما سربار پردازنده کم است زیرا فقط شماره‌دهی میشود.

**کشف بن‌بست و به عقب برگرداندن فرایندها:** کشف بن بست کار آسانی نیست. و محاسبات زیادی دارد به همین دلیلی میزان سربار پردازنده زیاد است. برای این کار علاوه بر محاسبه میبایست فهمید که فرایند باید به چه مرحله ای برگردد. که این موضوع و برگرداندن رجیسترها به همان حالت قبلی میزان سربار پردازنده را بالا میبرد ولی میزان موازی کار کردن فرایند ها را به محدود نمیکند.

ب) به طور کلی بله زیرا که اگر نرخ وقوع deadlock کم باشد الگوریتم هایی که به کشف deadlock و رفع آن میپردازند بهترند ولی اگر نرخ آن زیاد باشد استفاده از الگوریتم‌های پیش‌بینی بهترند.

_____تمرین ۳_____

در این روش **بن‌بست** ایجاد نمیشود اگر فیلسوفی بخواهد غذا بخورد و فقط یک چنگال در دسترس باشد آن را برنمیدارد به همین دلیل حتمن اگر بغل دستیش بخواهد بخورد میتواند غذا بخورد و به خاطر آن چنگال دچار وقفه نمیشود.و مطابق اصل لانه کبوتری بالاخره دو نفر از فیلسوفان غذا میخوند. قحطی‌زدگی وجود دارد. اگر بغل دستی های یک فرایند همیشه به سراغ چنگال ها بروند نوبت به آن فرد وسطی نمیرسد.و آن فرد وسطی موفق به غذا خوردن نمیشود و دچار گرسنگی میشود.

_____تمرین ۴_____

راهکاری برای این که به صورت قطعی تشخیص بدهیم که یک فرایند دچار گرسنگی شده است موجود نیست اما مثلا میتوان زمانی را اعلام کرد و گفت که اگر این زمان گذشت و به فرایند منابعش اختصاص داده نشد بگوییم دچار قحطی‌زدگی شده است.

میتوان برای رفع مشکل قحطی‌زدگی از راهکارهایی استفاده کرد:

- فرایندهایی زمان انتظارشان زیاد است در اولویتند.
- میتوان منبع را به فرایندی داد که اولویت اجراییش بیشتر است اما زمان انتظارش کمتر است به شرطی که فرایندهای دیگر دچار قحطی زدگی به همان صورت نشود.

_____تمرین ۵_____

اسم الگوریتم از مسئله ای در دنیای واقعی گرفته شده است. به نظر استفاده از ایده‌هایی که بیشترین سخت گیری را در زندگی دارد موجب میشود که زندگی دچار ایراد نشود ولی دچار روزمرگی میشود.

_____تمرین ٦_____

شباهت این دو موضوع متوقف شدن پیشرفت فرایندها بود. تفاوت‌هایش علت قفل شدن فرایند ها متفاوت است در deadlock یکی از فرایندها منبع بهش نمیرسد و هیچ کس منابع را رها نمیکند و معطل میماند. اما در livelock ایراد به این شکل است که فرایندها تلاش میکنند که کاری را انجام دهند اما موفق به انجام نمیشوند و معمولا وقتی اتفاق می‌افتد که دو فرایند همزمان به سراغ انجام یک کار میروند و هردو موفق به انجام نمیشوند. راهکار هم این است که فرایند ها صبر کنند و در زمانی تصادفی اقدام کنند به انجام کار.