به نام خدا

سوال ۱:

الف)

```
Condition x;
x.wait(c);
```

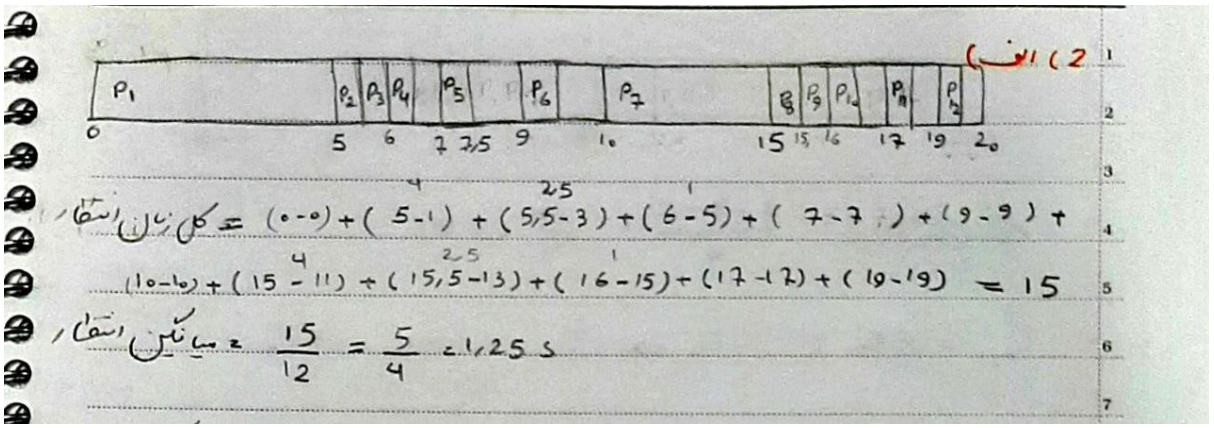
هنگامی که x.signal() صدا زده می شود با توجه به مقداری که در لحظه قفل کردن فرآیند یعنی x.wait(c) به c نسبت می دهیم، اولویت آن نسبت به بقیه فرآیندهای lock شده را تعیین می کند و هر فرآیند که بیشترین اولویت را داشته باشد، c کمتری داشته باشد، زودتر به ready queue برود.

ب)

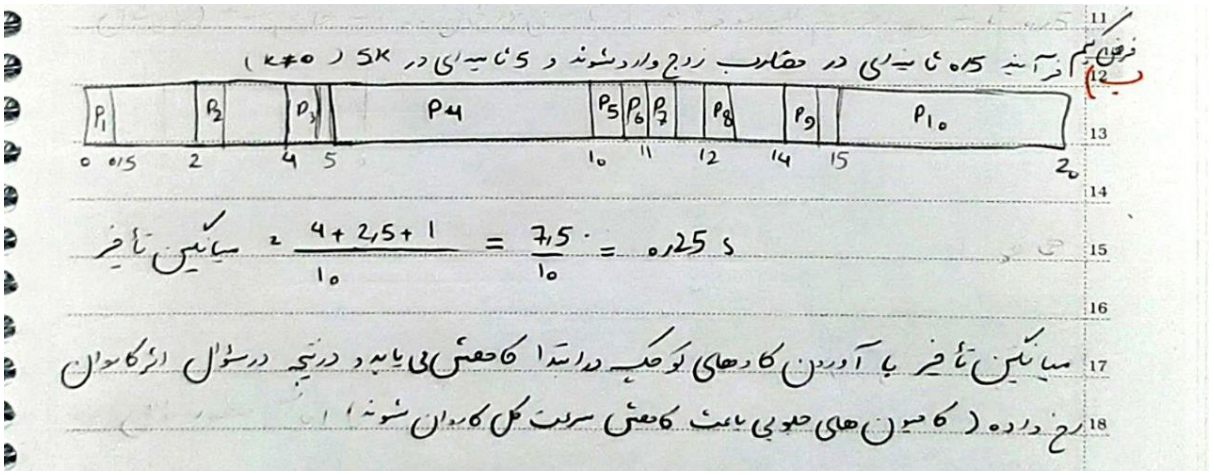
```
monitor Semaphore{
  struct Sem{
    int value,
    mutex lock,
    condition c
  };
  int bs_signal(Semaphore *sem){
    Lock(&sem->lock);
    sem->val++;
    if (sem->val <= 0) {
      signal(&sem->cond);
    }
    Unlock(&sem->lock);
    return 0;
  }
  int bc_wait(Semaphore *sem) {
    Lock(&sem->lock);
    if (sem->val-- <= 0) {
      wait(&sem->cond, &sem->lock);
    }
    unlock(&sem->lock);
  }
};
```

سوال ۲:

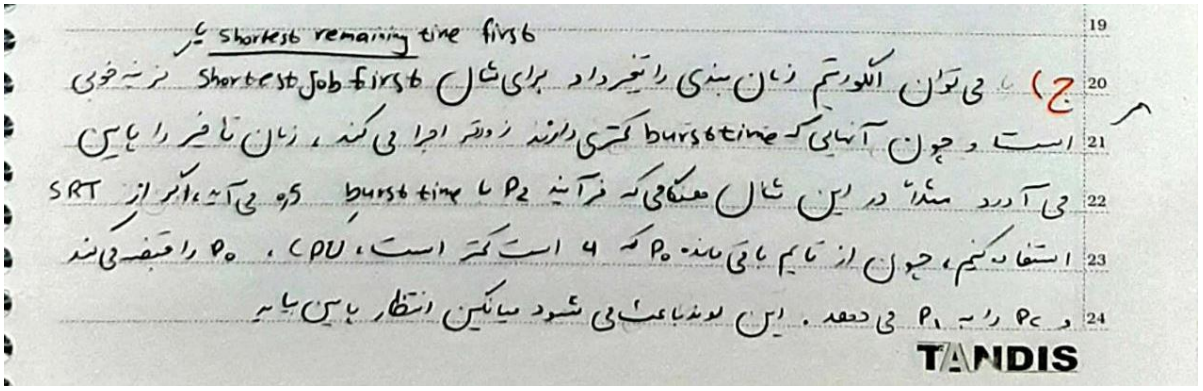
الف)



ب)

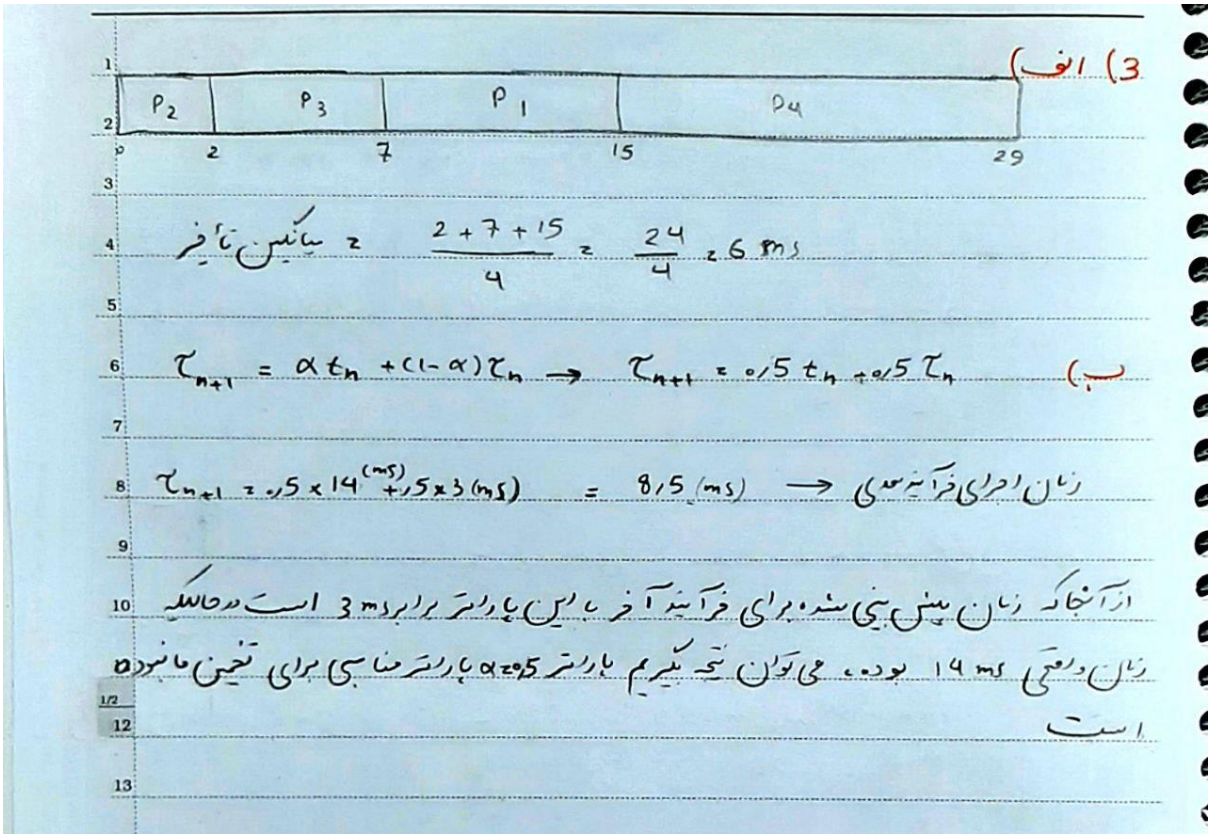


ج)



علاوه بر آن اگر امکان این وجود داشت که مدت زمان فرآیند ها را کاهش داد، میانگین زمان انتظار کاهش می یابد.

سوال ۳:



سوال ۴:

حافظه تراکنشی مدلی برای کنترل دسترسی های همزمان حافظه در محدوده برنامه نویسی موازی است. در برنامه نویسی موازی، تضمین می شود که فرآیندهایی که به صورت هم روند در حال اجرا هستند، منابع یکسان را به طور همزمان دستکاری نمی کنند. کنترل همزمان داده های حافظه مشترک از طریق قفل ها، به عنوان مثال، قفل های mutex انجام می شود. یک فرآیند قبل از تغییر داده های مشترک، قفل را می بندد و پس از آن قفل را آزاد می کند.

الف و ب) حافظه تراکنشی جایگزینی برای همگام سازی مبتنی بر قفل است. سعی می کند با گروه بندی عملیات خواندن و نوشتن و اجرای آنها مانند یک عملیات، برنامه نویسی موازی را ساده کند. حافظه تراکنشی مانند تراکنش های پایگاه داده است که در آن تمام دسترسی های حافظه مشترک و اثرات آنها یا همه با هم انجام می شود یا به صورت گروهی کنار گذاشته می شود(در واقع به صورت اتمیک انجام می شوند). همه فرآیندها می توانند به طور همزمان وارد منطقه بحرانی شوند. اگر در دسترسی به داده های حافظه مشترک تداخل وجود داشته باشد، فرآیند ها دوباره سعی می کنند به داده های حافظه مشترک دسترسی پیدا کنند یا بدون دستکاری داده های حافظه مشترک متوقف می شوند. بنابراین به حافظه تراکنشی، همگام سازی بدون قفل نیز می گویند. حافظه تراکنشی می تواند جایگزین رقابتی برای همگام سازی مبتنی بر قفل باشد.

ج) همگام‌سازی مبتنی بر قفل می‌تواند منجر به برخی مشکلات عملکرد شود زیرا فرآیندها ممکن است برای به‌روزرسانی داده‌های محصور شده (منطقه بحرانی) با قفل منتظر بمانند.