

## بسم تعالی

در زمان ورود پراسس جدید به صف یا خروج یک پراسس از صف (سررسید ددلاین یا پایان زمان سرویس) زمان باقیمانده از زمان سرویس سایر پراسس های در صف آپدیت می شود.

به این صورت که با فراخوانی تابع `updateEvents` در صورتی که یک ایونت `deadline` باشد تغییری نمی دهیم اما اگر `event` زمان پایان سرویس (`departure`) باشد زمان سر رسید آن را با توجه به شرایط جدید صف آپدیت خواهیم کرد.

```
def updateEvents(self):
    for i in range(len(self.events)):
        if self.events[i].type == "Deadline":
            continue
        new_service_time = self.processes[self.events[i].pid].service - (
            self.time - self.events[i].start_time) / self.events[i].queue_size
        self.processes[self.events[i].pid].service = new_service_time
        assert self.events[i].start_time <= self.time
        self.events[i].start_time = self.time
        self.events[i].queue_size = len(self.queue)
        self.events[i].time = self.time + new_service_time * len(self.queue)
    self.events.sort()
```

ابتدا `new_service_time` به صورت زیر محاسبه می شود:

$$\text{new service time} = \text{remaining service time} - \frac{\text{current time} - \text{event start time}}{\text{queue size}}$$

در نتیجه متناسب با اینکه چند پراسس دیگر در صف حضور داشته اند از زمان سرویس پراسس ها کم می شود و پس از آن مقادیر `start time` و `queue size` که در `event` ها ذخیره می شوند به زمان فعلی و سائز کنونی صف آپدیت می شوند تا در دفعات بعد مورد استفاده قرار گیرند و زمان سر رسید `event` نیز برابر با زمان کنونی به اضافه `new service time` خواهد بود که آن را آپدیت می کنیم.

در ضمن چون در صف `event` ها هر بار اولین `event` بررسی می شود نیاز است تا پس از تغییر زمان سرویس ها آن و آپدیت کردن زمان سر رسید `event` ها این صف را از نو `sort` کنیم.