



سیستم های عامل

Operating Systems

By:

Sanaz Shahræini

فصل سوم

الگوریتم های زمان بندی



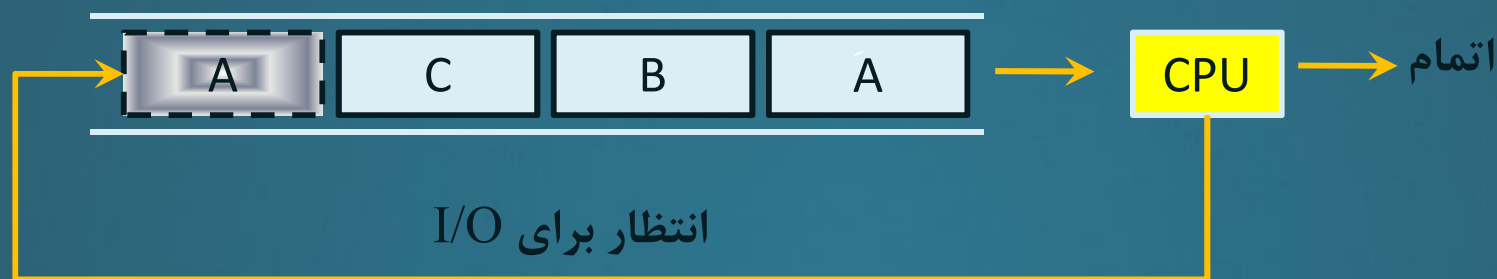
فهرست

۱. زمان بندی FCFS
۲. زمان بندی نوبت گردشی RR
۳. زمان بندی اولین کوتاه ترین کار SJF
۴. زمان بندی کوتاه ترین زمان باقیمانده SRT
۵. زمان بندی بالاترین نسبت پاسخ HRRN
۶. زمان بندی اولویت Priority
۷. زمان بندی شانسی Lottery
۸. زمان بندی اولین طولانی ترین کار LPT



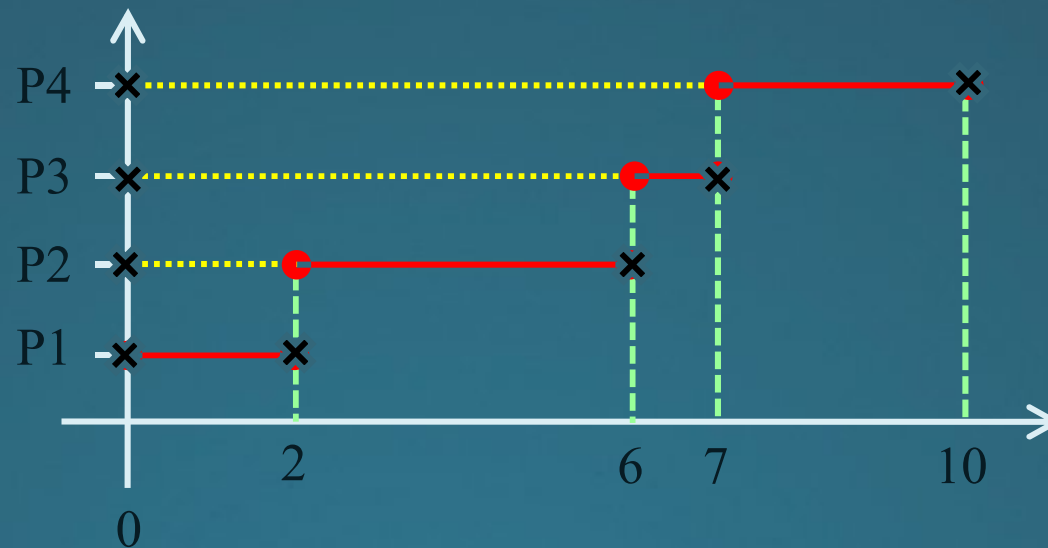
زمان بندی FCFS

- اولین ورودی - اولین سرویس گیرنده (First Come First Service) ساده ترین الگوریتم زمان بندی می باشد.
- در این روش هر فرآیندی که اولین درخواست را برای تصاحب CPU صادر کند، اولین فرآیندی خواهد بود که شروع به کار خواهد کرد.
- از دسته زمان بندی های **انحصاری** بوده و میانگین زمان انتظار آن غالباً طولانی است.
- این الگوریتم به سادگی با یک ساختمان داده صف پیاده سازی می شود.



مثال : در یک سیستم عامل از روش FCFS برای زمان بندی فرآیندها استفاده می شود. اگر فرآیندهای زیر در زمان صفر به ترتیب نام وارد سیستم شوند، آنگاه میانگین زمان انتظار را محاسبه کنید.

| فرآیندها | P1 | P2 | P3 | P4 |
|----------------|----|----|----|----|
| زمان اجرا (mS) | 2 | 4 | 1 | 3 |



$$\text{زمان انتظار} = \text{انتظار P1} + \text{انتظار P2} + \text{انتظار P3} + \text{انتظار P4} = 0 + 2 + 6 + 7 = 15 \text{ mS}$$

$$\text{میانگین زمان انتظار} = \frac{15}{4} = 3.75 \text{ mS}$$



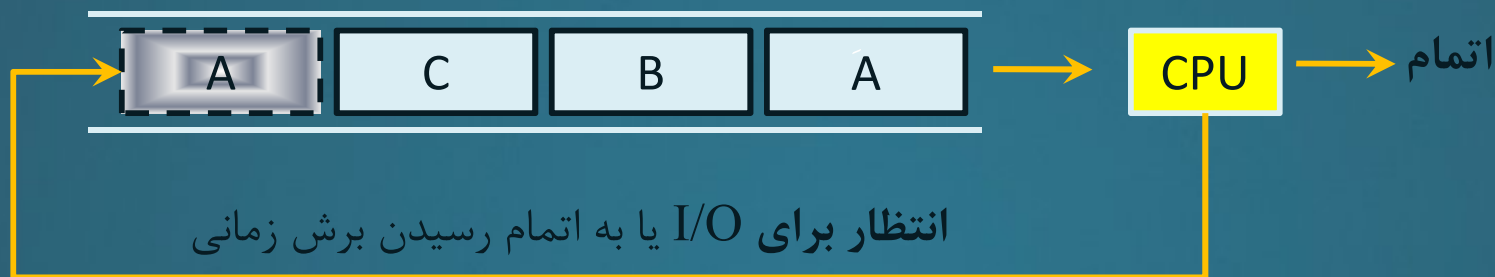
زمان بندی FCFS

- این الگوریتم اثر اسکورت (Covey Effect) را به دنبال دارد. یعنی اگر فرآیند اول تر زمان اجرای زیادی داشته باشد، فرآیند دارای زمان اجرای کوچکتر که بعد از آن قرار دارد مجبور است مدت زمان زیادی را منتظر فرآیند طولانی در صف بماند.
- این الگوریتم از این نظر که فرآیندها را به ترتیب ورود اجرا می کند عادلانه است، ولی در هنگام اثر اسکورت ناعادلانه می شود.
- از مزایای زمان بندی FCFS می توان به موارد زیر اشاره کرد :
 - سادگی پیاده سازی
 - اطمینان از اجرای فرآیند منتظر در صف آماده



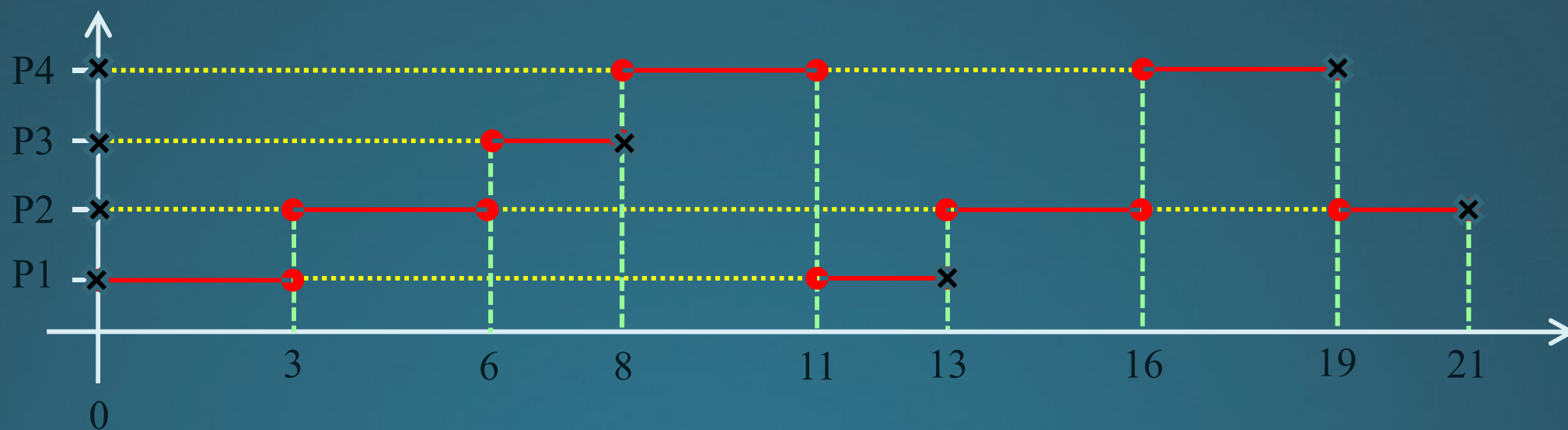
زمان بندی RR

- الگوریتم نوبت گردشی (Round Robin) از قدیمی ترین، ساده ترین، عادلانه ترین و رایج ترین الگوریتم های زمان بندی است.
- این روش مشابه FCFS بوده با این تفاوت که به هر فرآیند حداکثر به اندازه یک برش زمانی (Time Slice) که معمولاً بین ۱۰ تا ۱۰۰ میلی ثانیه است، اجازه استفاده از CPU را می دهد.
- میانگین زمان انتظار برای الگوریتم RR غالباً طولانی است.



مثال : در یک سیستم عامل از روش RR با برش زمانی 3 mS برای زمان بندی فرآیندها استفاده می شود. اگر فرآیندهای زیر در زمان صفر به ترتیب نام وارد سیستم شوند، آنگاه میانگین زمان انتظار، میانگین زمان پاسخ و میانگین زمان اجرا را محاسبه کنید.

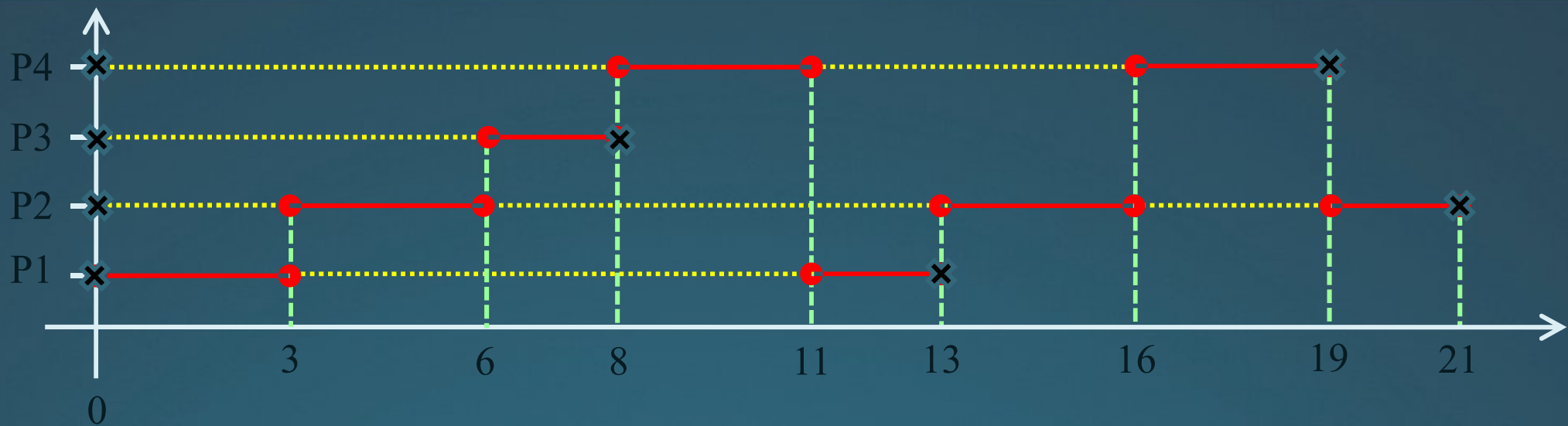
| فرآیندها | P1 | P2 | P3 | P4 |
|----------------|----|----|----|----|
| زمان اجرا (mS) | 5 | 8 | 2 | 6 |



$$\text{زمان اجرا} = \text{اجرا P1} + \text{اجرا P2} + \text{اجرا P3} + \text{اجرا P4} = 5 + 8 + 2 + 6 = 21 \text{ mS}$$

$$\text{میانگین زمان اجرا} = \frac{21}{4} = 5.25 \text{ mS}$$





$$\text{زمان انتظار} = P1 \text{ انتظار} + P2 \text{ انتظار} + P3 \text{ انتظار} + P4 \text{ انتظار} = 8 + 13 + 6 + 13 = 40 \text{ mS}$$

$$\text{میانگین زمان انتظار} = \frac{40}{4} = 10 \text{ mS}$$

$$\text{زمان پاسخ} = P1 \text{ پاسخ} + P2 \text{ پاسخ} + P3 \text{ پاسخ} + P4 \text{ پاسخ} = 13 + 21 + 8 + 19 = 61 \text{ mS}$$

$$\text{میانگین زمان پاسخ} = \frac{61}{4} = 15.25 \text{ mS}$$

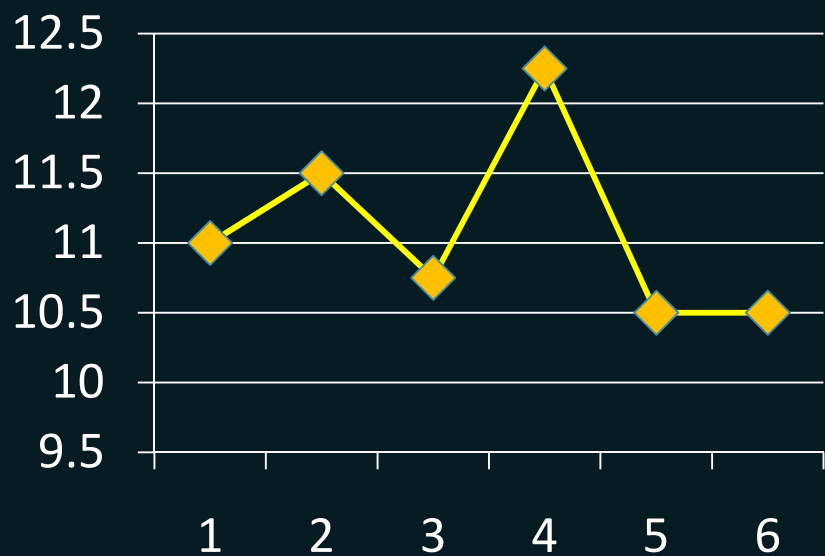
نکته! همانطور که در مثال قبلی مشاهده کردید، رابطه زیر برقرار است:

متوسط زمان انتظار + متوسط زمان اجرا = متوسط زمان پاسخ (برگشت)



زمان بندی RR

- کارآیی الگوریتم RR بطور کامل وابسته به مقدار برش زمانی است.
 - اگر برش زمانی بینهایت (بسیار بزرگ) در نظر گرفته شود، این الگوریتم شبیه الگوریتم FCFS خواهد شد.
 - اگر برش زمانی خیلی کوچک در نظر گرفته شود، بدلیل وقفه های زیاد بوجود آمده برای تعویض فرآیندها کارایی CPU بسیار کاهش پیدا خواهد کرد.



- میانگین زمان بازگشت یک مجموعه فرآیند الزاماً در اثر افزایش مقدار برش زمانی کاهش نمی یابد. به این مشکل **ناهنجاری بلیدی** می گویند.



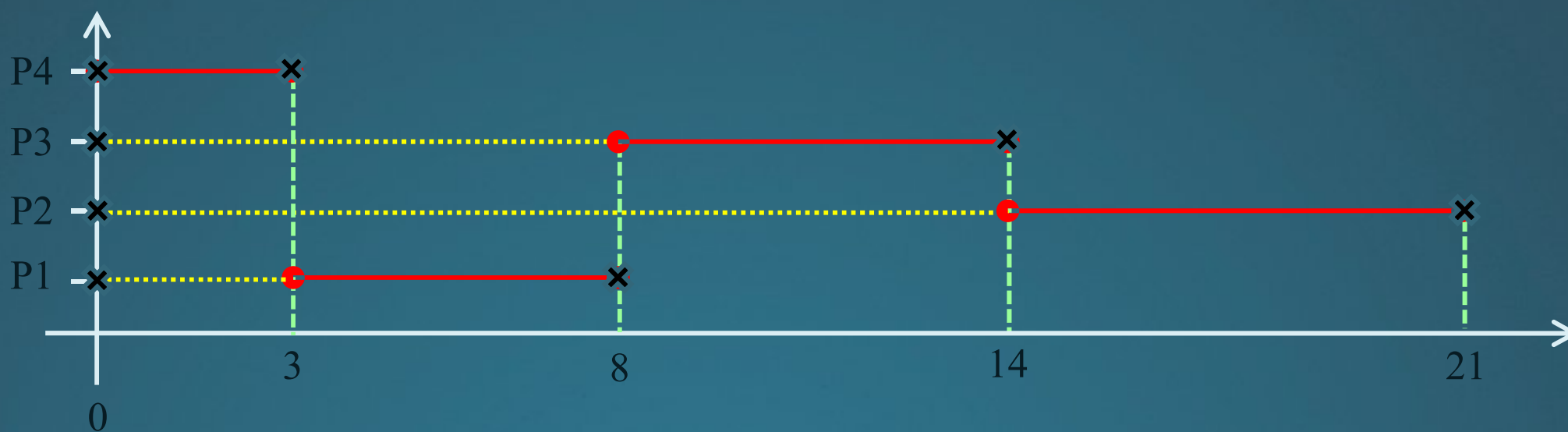
زمان بندی SJF

- اولین کوتاه ترین کار (Shortest Job First) الگوریتمی است که CPU را در اختیار فرآیندی قرار می دهد که کوچکترین زمان اجرا را خواهد داشت.
- این الگوریتم به نام SPN (Shortest Process Next) نیز مشهور است.
- نام مناسب تر برای این الگوریتم “کوتاهترین انفجار محاسباتی بعدی” نام دارد.
- این الگوریتم انحصاری عمل می کند و اگر دو فرآیند زمان اجرای یکسان داشته باشند، بر اساس الگوریتم FCFS زمان بندی می شوند.



مثال : در یک سیستم عامل از روش SJF برای زمان بندی فرآیندها استفاده می شود. اگر فرآیندهای زیر در زمان صفر وارد سیستم شوند، آنگاه میانگین زمان انتظار، میانگین زمان پاسخ و میانگین زمان اجرا را محاسبه کنید.

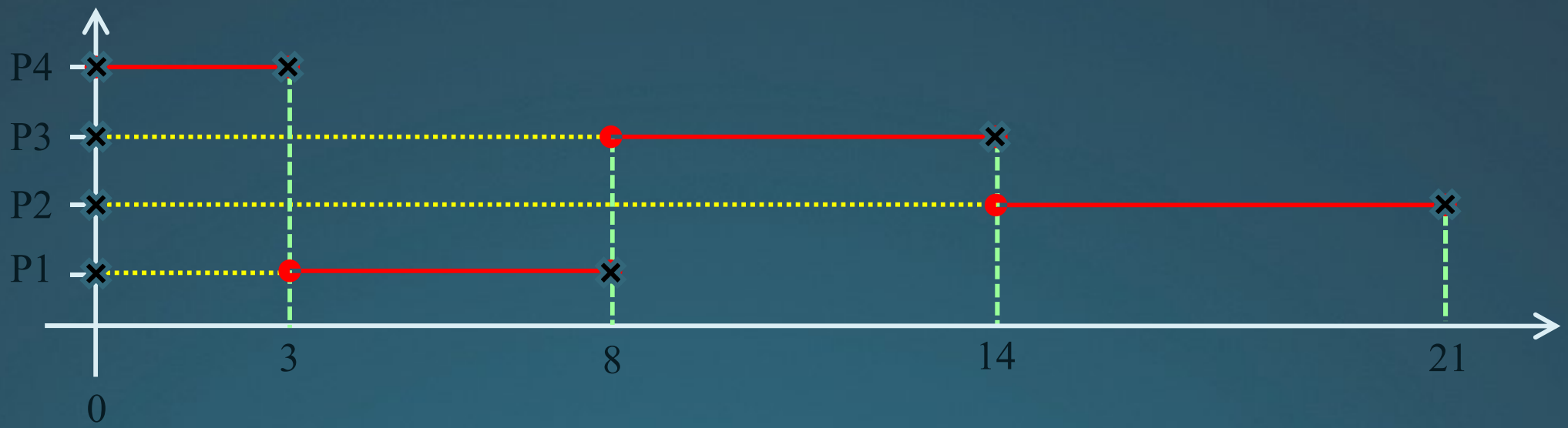
| فرآیندها | P1 | P2 | P3 | P4 |
|----------------|----|----|----|----|
| زمان اجرا (mS) | 5 | 7 | 6 | 3 |



$$\text{زمان اجرا} = \text{اجرا P1} + \text{اجرا P2} + \text{اجرا P3} + \text{اجرا P4} = 5 + 7 + 6 + 3 = 21 \text{ mS}$$

$$\text{میانگین زمان اجرا} = \frac{21}{4} = 5.25 \text{ mS}$$





$$\text{زمان انتظار} = P1 \text{ انتظار} + P2 \text{ انتظار} + P3 \text{ انتظار} + P4 \text{ انتظار} = 3 + 14 + 8 + 0 = 25 \text{ mS}$$

$$\text{میانگین زمان انتظار} = \frac{25}{4} = 6.25 \text{ mS}$$

$$\text{زمان پاسخ} = P1 \text{ پاسخ} + P2 \text{ پاسخ} + P3 \text{ پاسخ} + P4 \text{ پاسخ} = 8 + 21 + 14 + 3 = 46 \text{ mS}$$

$$\text{میانگین زمان پاسخ} = \frac{46}{4} = 11.5 \text{ mS}$$

$$\text{میانگین زمان پاسخ} = \text{میانگین زمان اجرا} + \text{میانگین زمان انتظار} = 5.25 + 6.25 = 11.5 \text{ mS}$$



زمان بندی SJF

- در حالت کلی ثابت می شود که الگوریتم SJF دارای بهینه ترین (کمترین) میانگین زمان انتظار یا کمترین میانگین زمان بازگشت در میان الگوریتم های انحصاری می باشد.
- این الگوریتم مخصوصاً برای کارهای دسته ای که زمان اجرای آنها از قبل مشخص است بکار می رود.
- مشکلات این الگوریتم عبارتند از :
 - آگاهی از مدت زمان مورد نیاز فرآیند بعدی (پیش بینی زمان اجرای فرآیند بعدی)
 - امکان گرسنگی یا قحطی فرآیندهای دارای زمان اجرای طولانی



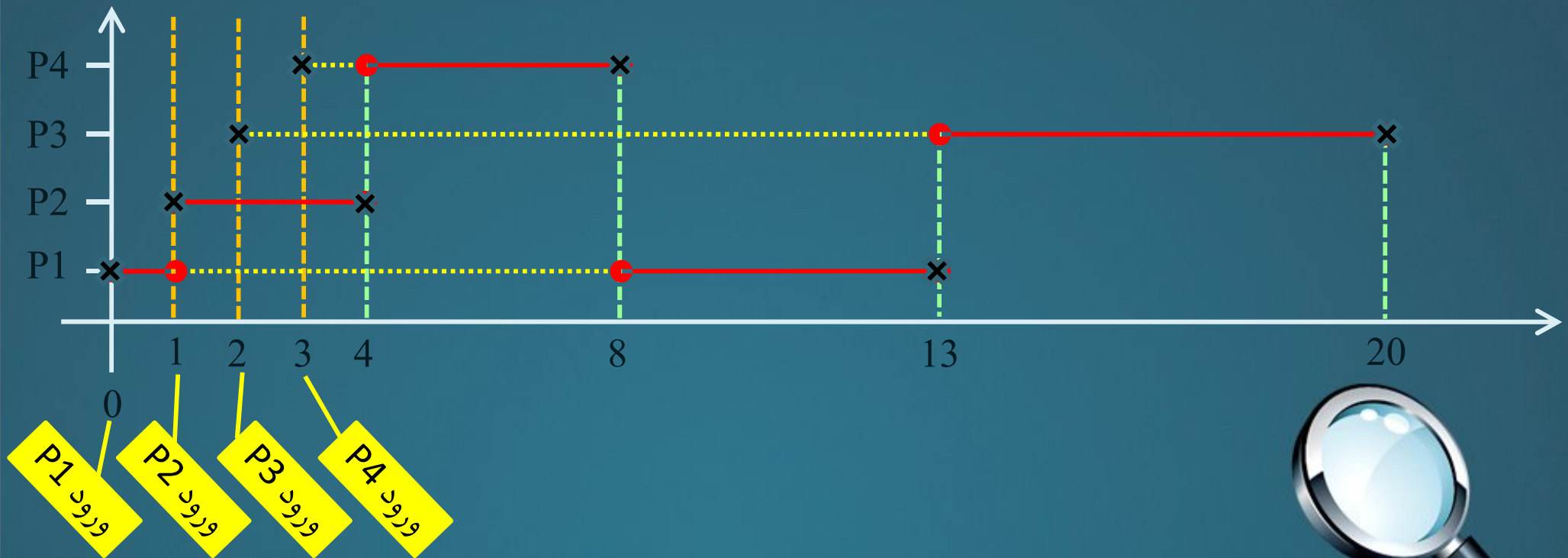
زمان بندی SRT

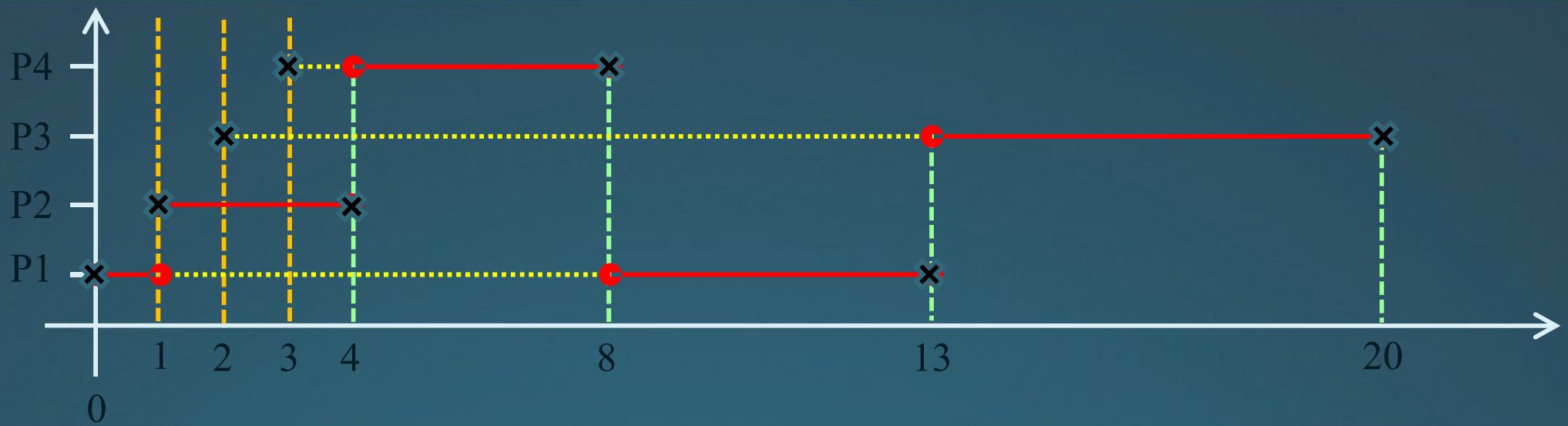
- کوتاه ترین زمان باقیمانده (Shortest Remaining Time) فرآیندی را جلوتر اجرا خواهد کرد که به کمترین زمان جهت تکمیل شدن احتیاج دارد.
- نام دیگر این الگوریتم SRTF (Shortest Remaining Time First) می باشد.
- این الگوریتم شبیه SJF می باشد ولی بصورت غیر انحصاری می باشد.
- در این روش فرآیندها الزاماً در یک زمان و باهم وارد نمی شوند. به عبارت دیگر فرآیندهای مختلف در زمان های مختلف می توانند برای اجرا درخواست صادر کنند.



مثال : در یک سیستم عامل از روش SRT برای زمان بندی فرآیندها استفاده می شود. اگر فرآیندهای زیر وارد سیستم شوند، آنگاه میانگین زمان انتظار، میانگین زمان پاسخ و میانگین زمان اجرا را محاسبه کنید و آنها را با روش SJF مقایسه کنید.

| فرآیندها | P1 | P2 | P3 | P4 |
|----------------|----|----|----|----|
| زمان اجرا (mS) | 6 | 3 | 7 | 4 |
| زمان ورود (mS) | 0 | 1 | 2 | 3 |





$$\text{زمان اجرا} = \text{P1 اجرا} + \text{P2 اجرا} + \text{P3 اجرا} + \text{P4 اجرا} = 6 + 3 + 7 + 4 = 20 \text{ mS}$$

$$\text{میانگین زمان اجرا} = \frac{20}{4} = 5 \text{ mS}$$

$$\text{زمان انتظار} = \text{P1 انتظار} + \text{P2 انتظار} + \text{P3 انتظار} + \text{P4 انتظار} = 7 + 0 + 11 + 1 = 19 \text{ mS}$$

$$\text{میانگین زمان انتظار} = \frac{19}{4} = 4.75 \text{ mS}$$

$$\text{زمان پاسخ} = \text{P1 پاسخ} + \text{P2 پاسخ} + \text{P3 پاسخ} + \text{P4 پاسخ} = 13 + 3 + 18 + 5 = 39 \text{ mS}$$

$$\text{میانگین زمان پاسخ} = \frac{39}{4} = 9.75 \text{ mS}$$

$$\text{SJF میانگین زمان اجرا} = 5 \text{ mS}$$

$$\text{SJF میانگین زمان انتظار} = 5.5 \text{ mS}$$

$$\text{SJF میانگین زمان پاسخ} = 10.5 \text{ mS}$$



زمان بندی HRRN

- بالاترین نسبت پاسخ (Highest Response Ratio Next) الگوریتمی است که بصورت پویا اولویت اجرای فرآیند ها را از رابطه زیر محاسبه می کند:

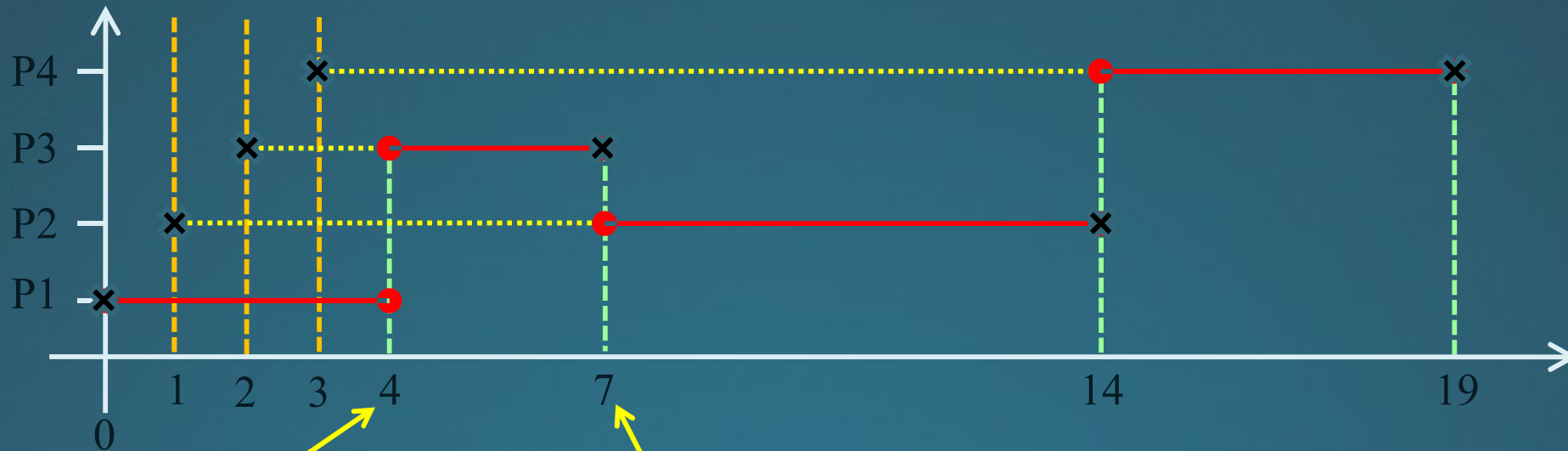
$$\text{اولویت} = \frac{\text{زمان سرویس دهی} + \text{زمان انتظار}}{\text{زمان سرویس دهی}}$$

- این الگوریتم توسط Brinch Hansen ابداع شده و بعضی از مشکلات الگوریتم SJF را برطرف می کند.
- عملکرد این الگوریتم بصورت انحصاری می باشد.



مثال : در یک سیستم عامل روش HRRN در زمان بندی فرآیندها زیر استفاده می شود.
 میانگین زمان انتظار، میانگین زمان پاسخ و میانگین زمان اجرا را محاسبه کنید.

| فرآیندها | P1 | P2 | P3 | P4 |
|----------------|----|----|----|----|
| زمان اجرا (mS) | 4 | 7 | 3 | 5 |
| زمان ورود (mS) | 0 | 1 | 2 | 3 |



$$P2 = \frac{3 + 7}{7} = 1.43$$

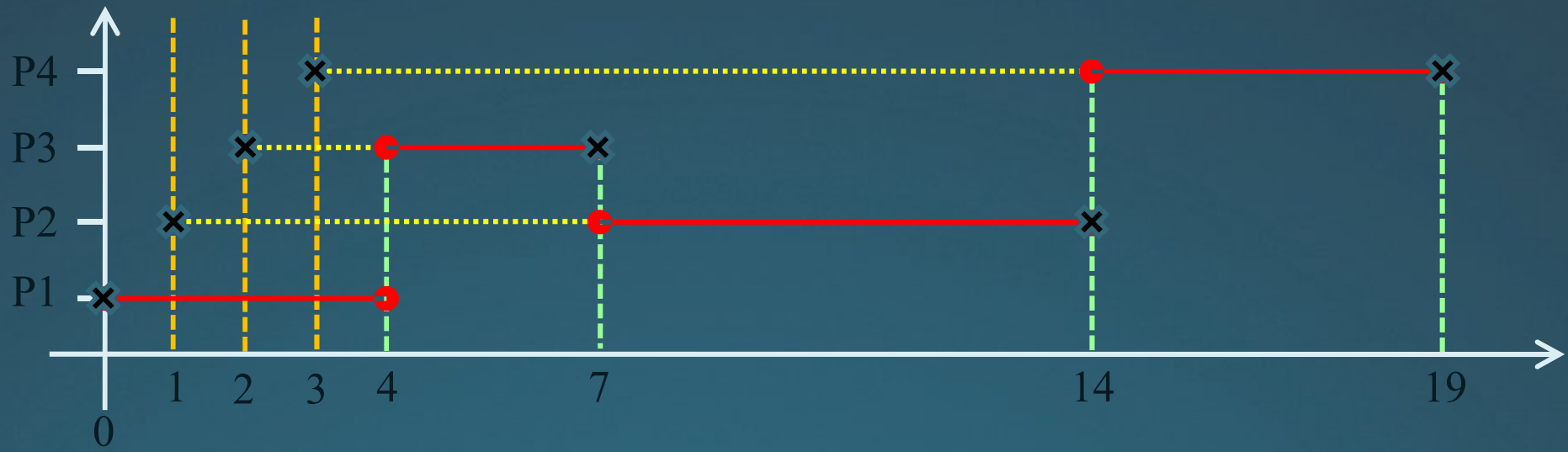
$$P3 = \frac{2 + 3}{3} = 1.66$$

$$P4 = \frac{1 + 5}{5} = 1.2$$

$$P2 = \frac{6 + 7}{7} = 1.86$$

$$P4 = \frac{4 + 5}{5} = 1.8$$





$$\text{زمان اجرا} = \text{P1 اجرا} + \text{P2 اجرا} + \text{P3 اجرا} + \text{P4 اجرا} = 4 + 7 + 3 + 5 = 19 \text{ mS}$$

$$\text{میانگین زمان اجرا} = \frac{19}{4} = 4.75 \text{ mS}$$

$$\text{زمان انتظار} = \text{P1 انتظار} + \text{P2 انتظار} + \text{P3 انتظار} + \text{P4 انتظار} = 0 + 6 + 2 + 11 = 19 \text{ mS}$$

$$\text{میانگین زمان انتظار} = \frac{19}{4} = 4.75 \text{ mS}$$

$$\text{زمان پاسخ} = \text{P1 پاسخ} + \text{P2 پاسخ} + \text{P3 پاسخ} + \text{P4 پاسخ} = 4 + 13 + 5 + 16 = 38 \text{ mS}$$

$$\text{میانگین زمان پاسخ} = \frac{38}{4} = 9.5 \text{ mS}$$



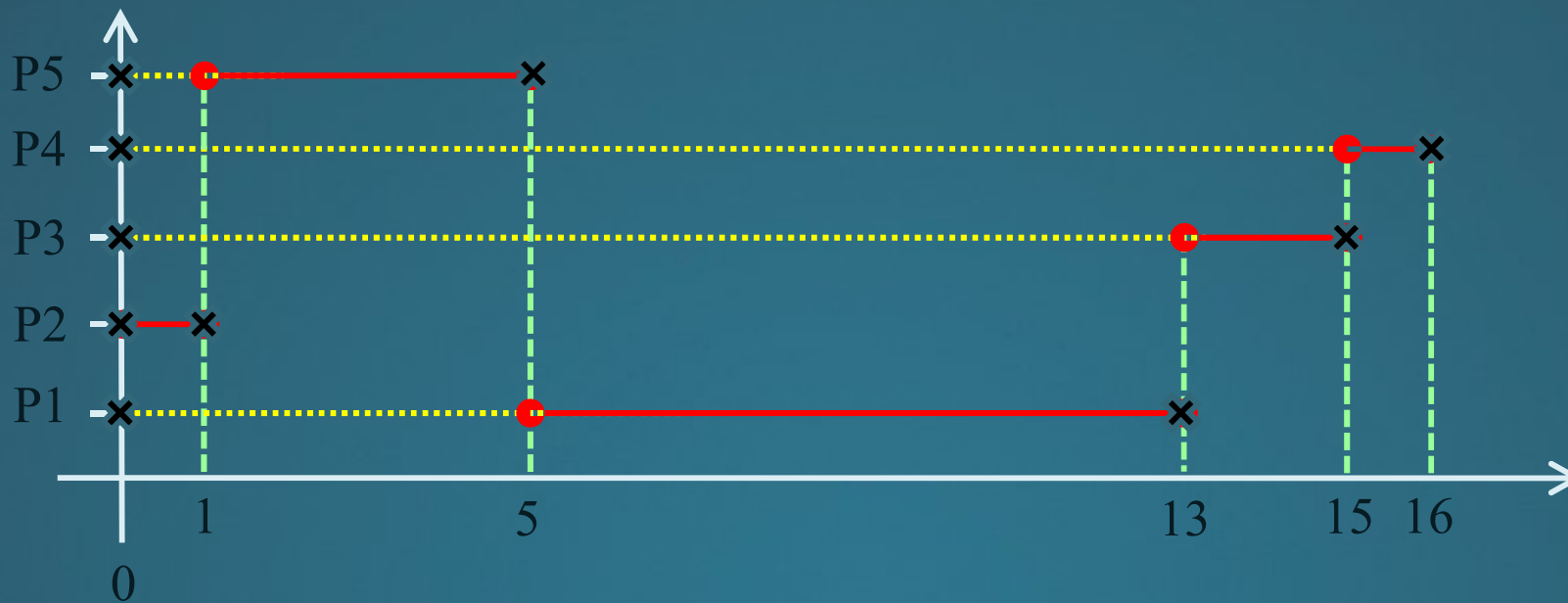
زمان بندی اولویت

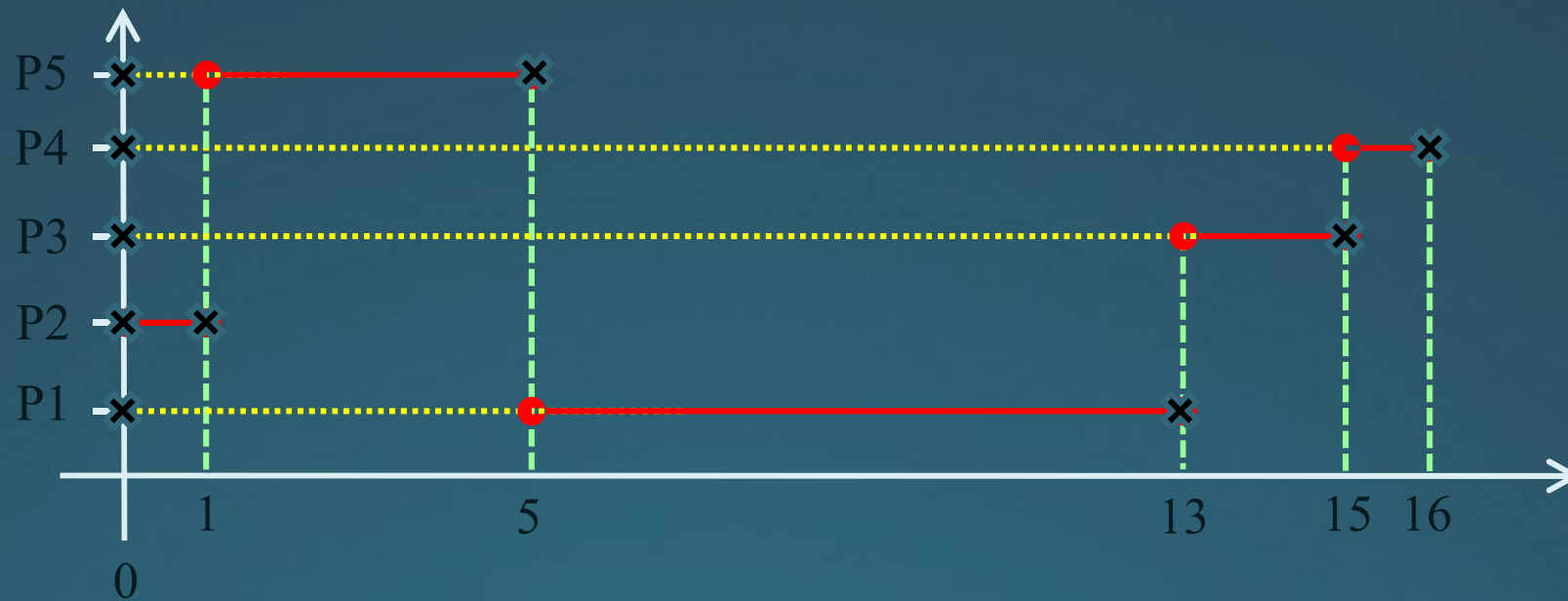
- الگوریتم اولویت (Priority) بر اساس اولویت تعیین شده توسط کاربر یا سیستم عامل، فرآیندها را زمان بندی می کند.
- می تواند هم انحصاری و هم غیر انحصاری پیاده سازی شود. ولی اغلب بصورت انحصاری مورد استفاده قرار می گیرد.
- اصلی ترین مشکل این الگوریتم، گرسنگی یا قحطی فرآیندهای با اولویت کم می باشد.
- در واقع الگوریتم های SJF و HRRN نوعی الگوریتم اولویت هستند.
- در سیستم عامل های UNIX و Windows NT از ترکیبی از الگوریتم های اولویت و RR استفاده می شود.



مثال : در یک سیستم عامل از روش اولویت برای زمان بندی فرآیندها استفاده می شود. اگر فرآیندهای زیر با هم از زمان صفر وارد سیستم شوند، آنگاه میانگین زمان انتظار، میانگین زمان پاسخ و میانگین زمان اجرا را محاسبه کنید.

| فرآیندها | P1 | P2 | P3 | P4 | P5 |
|----------------|----|----|----|----|----|
| زمان اجرا (mS) | 8 | 1 | 2 | 1 | 4 |
| اولویت اجرا | 3 | 1 | 3 | 4 | 2 |





$$\text{زمان اجرا} = \text{P1 اجرا} + \text{P2 اجرا} + \text{P3 اجرا} + \text{P4 اجرا} + \text{P5 اجرا} = 8 + 1 + 2 + 1 + 4 = 16 \text{ mS}$$

$$\text{میانگین زمان اجرا} = \frac{16}{5} = 3.2 \text{ mS}$$

$$\text{زمان انتظار} = \text{P1 انتظار} + \text{P2 انتظار} + \text{P3 انتظار} + \text{P4 انتظار} + \text{P5 انتظار} = 5 + 0 + 13 + 15 + 1 = 34 \text{ mS}$$

$$\text{میانگین زمان انتظار} = \frac{34}{5} = 6.8 \text{ mS}$$

$$\begin{aligned} \text{زمان پاسخ} &= \text{P1 پاسخ} + \text{P2 پاسخ} + \text{P3 پاسخ} + \text{P4 پاسخ} + \text{P5 پاسخ} \\ &= 13 + 1 + 15 + 16 + 5 = 50 \text{ mS} \end{aligned}$$

$$\text{میانگین زمان پاسخ} = \frac{50}{5} = 10 \text{ mS}$$



زمان بندی شانس

- الگوریتم **شانسی (Lottery)** بر اساس شانس (قرعه کشی) فرآیندها را اجرا می کند.
- در این الگوریتم، سیستم عامل به هر فرآیند تعدادی عدد اختصاص می دهد که تعداد اعداد اختصاص داده شده بر اساس اولویت هر فرآیند می باشد (هر چه اولویت فرآیند بالاتر باشد تعداد اعداد فرآیند بیشتر خواهد بود). سپس سیستم بطور اتفاقی یک عدد تولید می کند و فرآیندی که آن عدد را در اختیار دارد اجرا خواهد شد.
- این الگوریتم بصورت **انحصاری** می باشد.



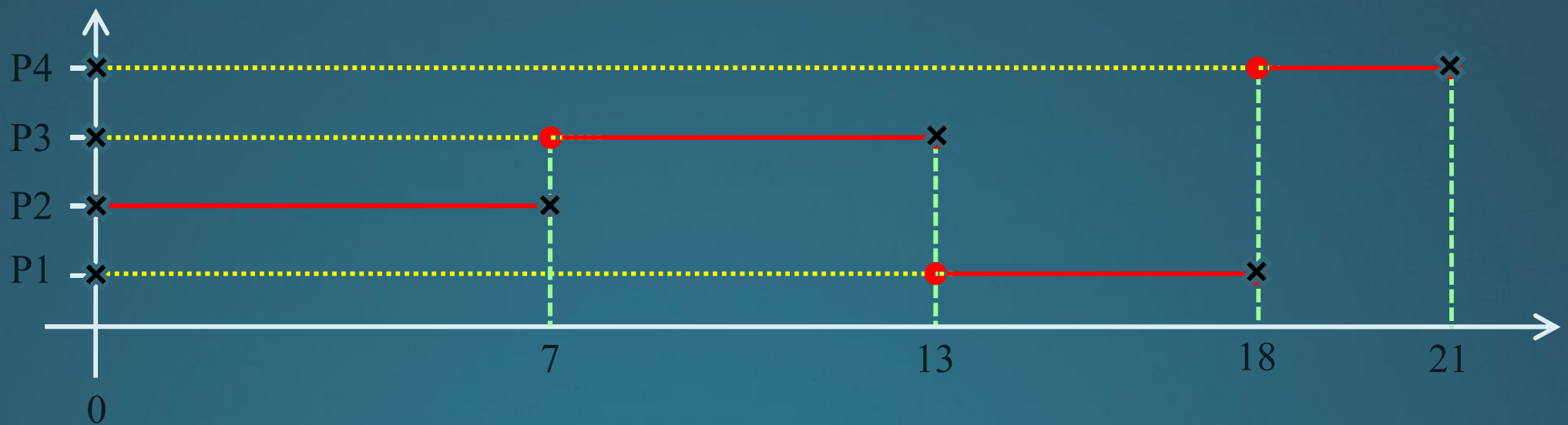
زمان بندی LPT

- بلندترین زمان اجرا (Longest Processing Time) از بین فرآیندهای موجود، فرآیندی را برای اجرا انتخاب می کند که طولانی ترین زمان اجرا را داشته باشد.
- این الگوریتم دقیقاً برعکس الگوریتم SJF عمل می کند.
- این الگوریتم اغلب بصورت **انحصاری** اجرا می شود.
- اغلب برای سیستم های چند پردازنده ای از این الگوریتم استفاده می شود.



مثال : در یک سیستم عامل از روش LPT برای زمان بندی فرآیندها استفاده می شود. اگر فرآیندهای زیر در زمان صفر وارد سیستم شوند، آنگاه میانگین زمان انتظار، میانگین زمان پاسخ و میانگین زمان اجرا را محاسبه کنید و نتایج را با الگوریتم SJF مقایسه کنید.

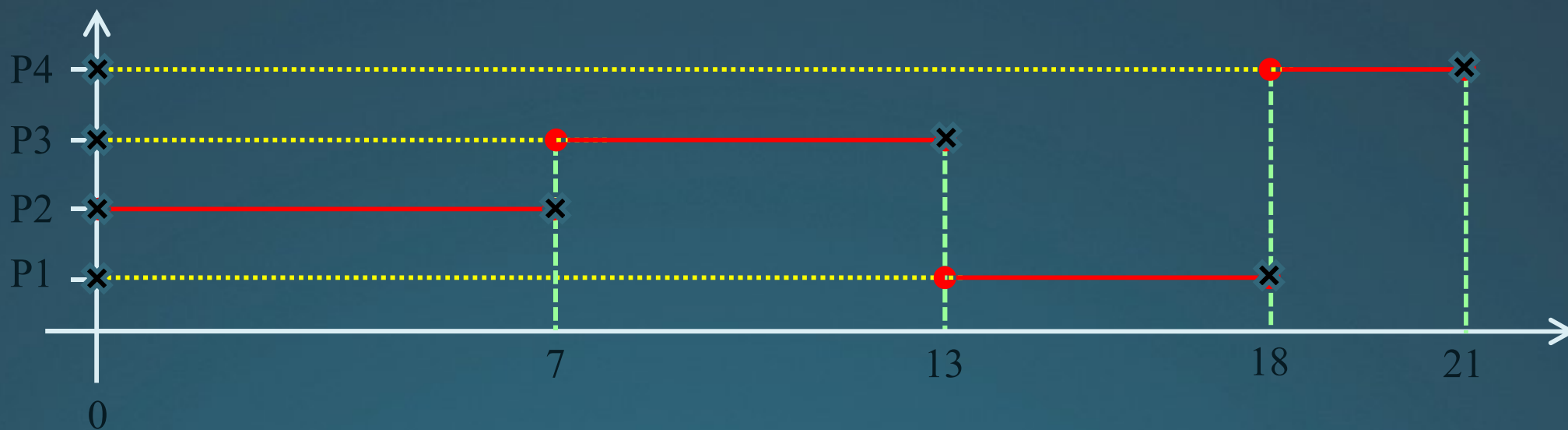
| فرآیندها | P1 | P2 | P3 | P4 |
|----------------|----|----|----|----|
| زمان اجرا (mS) | 5 | 7 | 6 | 3 |



$$\text{زمان اجرا} = \text{اجرا P1} + \text{اجرا P2} + \text{اجرا P3} + \text{اجرا P4} = 5 + 7 + 6 + 3 = 21 \text{ mS}$$

$$\text{میانگین زمان اجرا} = \frac{21}{4} = 5.25 \text{ mS}$$





$$\text{زمان انتظار} = P1 \text{ انتظار} + P2 \text{ انتظار} + P3 \text{ انتظار} + P4 \text{ انتظار} = 13 + 0 + 7 + 18 = 38 \text{ mS}$$

$$\text{میانگین زمان انتظار} = \frac{38}{4} = 9.5 \text{ mS}$$

$$\text{زمان پاسخ} = P1 \text{ پاسخ} + P2 \text{ پاسخ} + P3 \text{ پاسخ} + P4 \text{ پاسخ} = 18 + 7 + 13 + 21 = 59 \text{ mS}$$

$$\text{میانگین زمان پاسخ} = \frac{59}{4} = 14.75 \text{ mS}$$

$$\text{میانگین زمان پاسخ} = \text{میانگین زمان اجرا} + \text{میانگین زمان انتظار} = 5.25 + 9.5 = 14.75 \text{ mS}$$

$$\text{SJF میانگین زمان اجرا} = 5.25 \text{ mS}$$

$$\text{SJF میانگین زمان انتظار} = 6.25 \text{ mS}$$

$$\text{SJF میانگین زمان پاسخ} = 11.5 \text{ mS}$$



سوال ؟

