

فصل چهارم

Linux

Apple



Windows

الگوریتمهای زمانبندی CPU

الگوریتمهای زمانبندی CPU عبارتند از: ۱- FCFS ۲- Round Robin ۳- SJF ۴- SRT ۵- HRRN ۶- Priority S. ۷- Multi Queue S.

۸- Multi Queue & Feedback S. ۹- Chance S. ۱۰- LPT

FCFS

ساده ترین نوع زمانبندی می باشد. اولین پردازش که به صف وارد شود، اول خدمات می گیرد. (First Come First Service)

یا (First In First Out) نامیده می شود.

از نوع انحصاری می باشد (non-preemptive).

نام پردازش	P1	P2	P3
زمان انفجار	24	3	3

سه پردازش همزمان با زمانهای انفجار CPU وارد شده اند، میانگین زمان انتظار را محاسبه کنید.



$$\text{میانگین زمان انتظار} = (0 + 24 + 27) / 3 = 17$$

همانطور که دیده می شود اگر یک پردازش CPU limited با زمان زیاد باشد بقیه پردازشهای با زمان پردازش کم، دچار انتظار زیاد می شوند.

(اثر اسکورت Convoy effect). در نتیجه این زمانبندی برای کارهای محاوره ای مفید نیست و غیر عادلانه می باشد. سادگی و اینکه تمام پردازشهای

درون صف بالاخره اجرا خواهند شد از مزایای آن می باشد.

Round Robin RR

پردازشها داخل صف هستند مانند FCFS اما این روش غیر انحصاری (Preemptive) می باشد و یک برش زمانی به نام کوانتوم (Time Quantum) دارد. پردازشی که بیش از این زمان، CPU را اشغال کند به انتهای صف هدایت می شود. برای سیستمهای اشتراک زمانی و محاوره ای (Interactive) می باشد.

زمان انتظار + زمان اجرا = زمان پاسخ یا برگشت

یا می توان گفت

زمان ورود پردازش - زمان خروج پردازش = زمان پاسخ یا برگشت

زمان در صف بودن = زمان انتظار

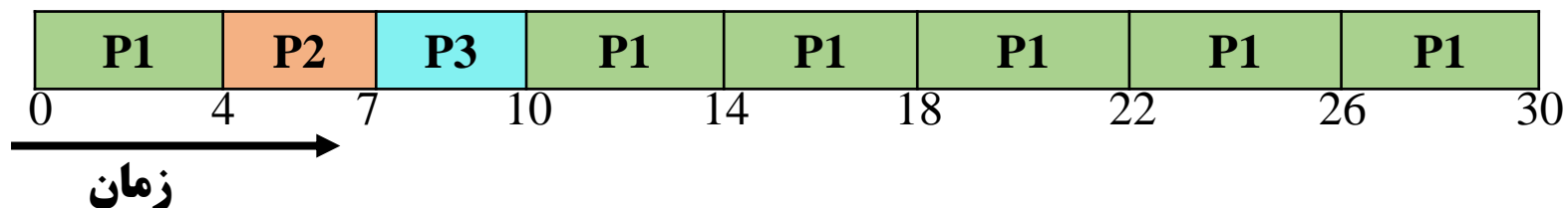
یا می توان گفت

زمان اجرا - زمان ورود - زمان خروج = زمان انتظار

زمان اشغال cpu = زمان اجرا

سه پردازش همزمان (در زمان صفر)، با زمانهای انفجار CPU وارد شده اند، زمان **کوانتو ۴ است**، میانگین زمان انتظار، پاسخ و اجرا را محاسبه کنید.

نام پردازش	P1	P2	P3
زمان انفجار	24	3	3



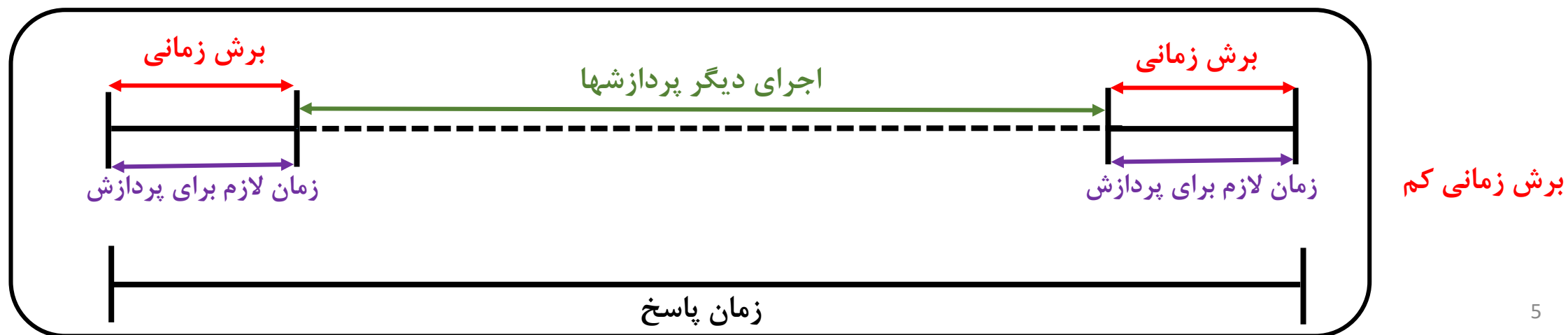
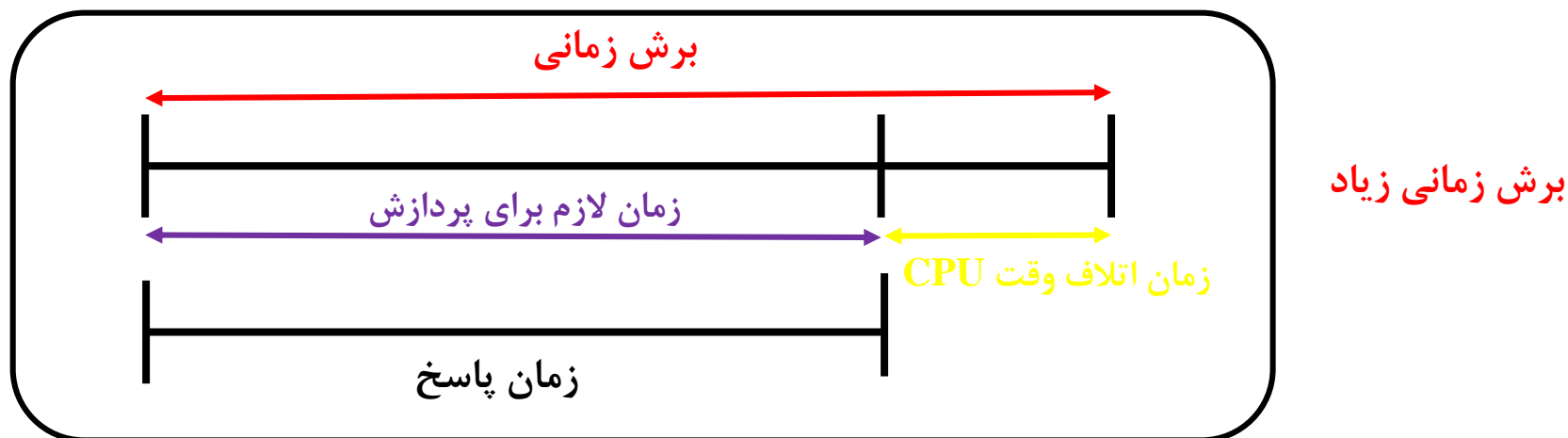
$$\text{میانگین زمان پاسخ (۲)} = \frac{30 + (7 - 0) + (10 - 0)}{3} = \frac{47}{3} = 15.66$$

$$\text{میانگین زمان انتظار (۱)} = (6 + 4 + 7) / 3 = 5.66$$

$$\text{میانگین زمان اجرا (۳)} = \frac{3 + 3 + 24}{3} = \frac{30}{3} = 10$$

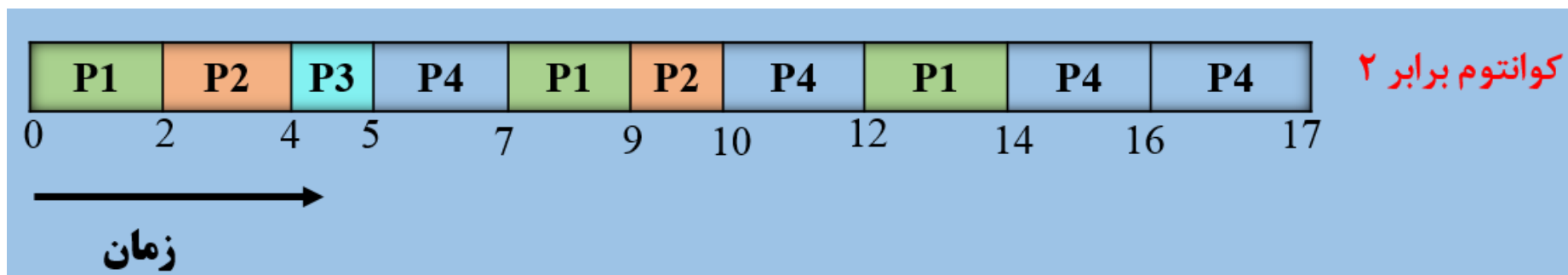
- ❑ مقدار کوانتوم اگر خیلی کم باشد، تعدا تعویض متن بالا می رود. چون در زمان تعویض متن CPU عمل مفیدی انجام نمی دهد. بنا براین باید زمان کوانتوم هزاران بار از زمان تعویض متن بیشتر باشد. اگر زمان کوانتوم بی نهایت بشود. زمانبندی RR تبدیل به FCFS می شود.
- از فرمول مقابل برای یافتن زمان مناسب کوانتوم معمولاً استفاده می شود:
- ❑ اثر مقدار خیلی کم یا خیلی زیاد بودن مقدار کوانتوم بر زمان پاسخ در نمودار زیر دیده می شود.

$$\text{برش زمانی} \times N = \text{زمان پاسخ مناسب}$$



چهارپردازش همزمان یا در زمان صفر با زمانهای انفجار CPU داده شده وارد شده اند، میانگین زمان برگشت را برای کوانتوم های ۱ تا ۷ محاسبه کرده و نمودار میانگین زمان برگشت به کوانتوم را رسم کنید.

نام پردازش	P1	P2	P3	P4
زمان انفجار CPU	6	3	1	7



۱) میانگین زمان انتظار

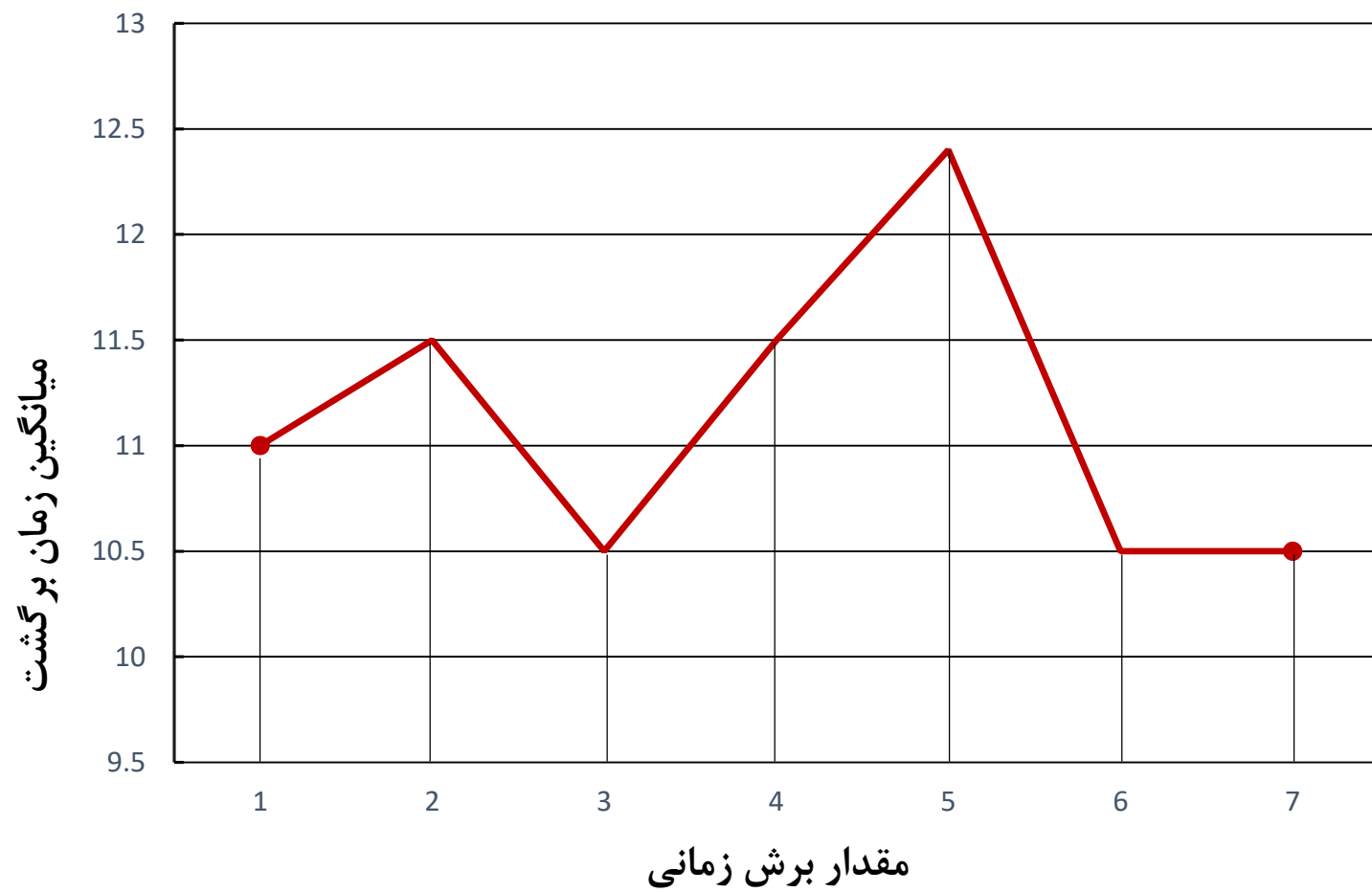
$$= \frac{(14-6)+(10-3)+(5-1)+(17-7)}{4} = \frac{8+7+4+10}{4} = \frac{29}{4} = 7.25$$

۲) میانگین زمان پاسخ

$$= \frac{14+10+5+17}{4} = \frac{46}{4} = 11.5$$

۳) میانگین زمان اجرا

$$= \frac{7+1+3+6}{4} = \frac{17}{4} = 4.25$$



Shortest Job First SJF

□ در این روش CPU به پردازشی که کمترین زمان انفجار محاسباتی را دارد اختصاص داده می شود.

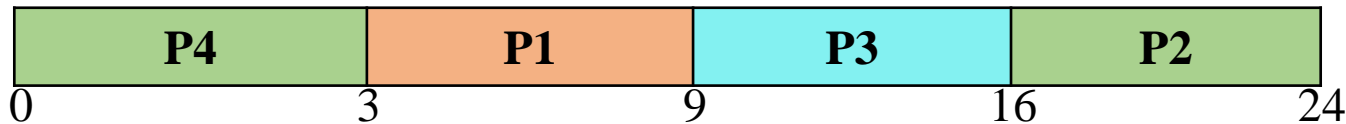
روشی **انحصاری** می باشد و اگر دو پردازش دارای زمان انفجار محاسباتی یکسان باشند،

آنگاه به روش FCFS رفتار می شود.

میانگین زمان انتظار را برای پردازشهای زیر در دو الگوریتم SJF و FCFS محاسبه کنید

نام پردازش	P1	P2	P3	P4
زمان انفجار CPU	6	8	7	3

نمودار گانت SJF



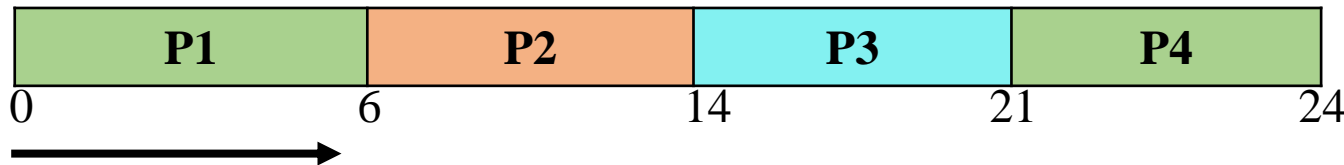
زمان

$$\text{میانگین زمان انتظار} = \frac{0 + 3 + 9 + 16}{4} = \frac{28}{4} = 7$$

این الگوریتم بهترین زمان بازگشت در بین الگوریتمهای انحصاری دارد.

امکان قحطی Starvation برای پردازشهای با زمان طولانی دارد.

نمودار گانت FCFS



زمان

$$\text{میانگین زمان انتظار} = \frac{0 + 6 + 14 + 21}{4} = \frac{41}{4} = 10.25$$

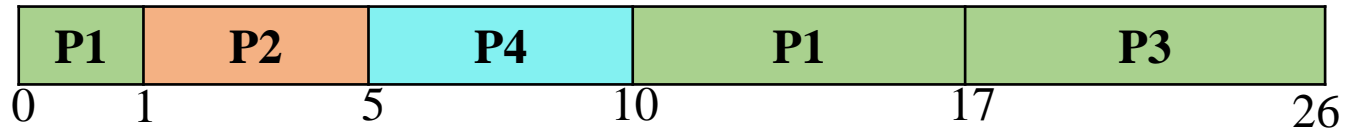
Shortest Remaining Time SRT

□ در این روش ابتدا CPU به پردازش اول داده می شود، اگر پردازشی با زمان اجرای کمتر رسید، جای آن را می گیرد. در نتیجه الگوریتم **غیر انحصاری** می باشد. در سیستمهای اشتراک زمانی کاربرد دارد. احتمال ایجاد پدیده گرسنگی وجود دارد. هزینه تعویض متن بیشتری دارد.

نام پردازش	P1	P2	P3	P4
زمان انفجار CPU	8	4	9	5
زمان ورود	0	1	2	3

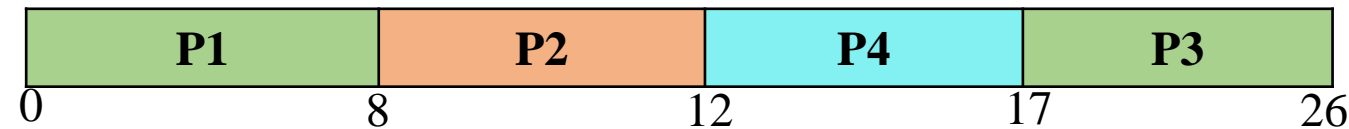
❖ میانگین زمان انتظار را برای پردازشهای زیر در دو الگوریتم SJF و SRT محاسبه کنید

نمودار گانت SRT



$$\text{میانگین زمان انتظار} = \frac{(10 - 1) + (1 - 1) + (17 - 2) + (5 - 3)}{4} = \frac{26}{4} = 6.5$$

نمودار گانت SJF



$$\text{میانگین زمان انتظار} = \frac{(0 - 0) + (8 - 1) + (12 - 3) + (17 - 2)}{4} = \frac{28}{4} = 7.75$$

Highest Response Ratio Next HRRN /HRN

□ زمانبندی از نوع **انحصاری** می باشد. جهت تصحیح روش SJF ابتدا شده است. در روش SJF تاکید روی کارهای کوتاهتر است؛ این الگوریتم آن را تصحیح می کند تا کارهای بلند دچار قحطی زدگی نشوند. هر کاری که الویت بیشتری دارد، ابتدا اجرا می شود. الویت از فرمول زیر محاسبه می شود:

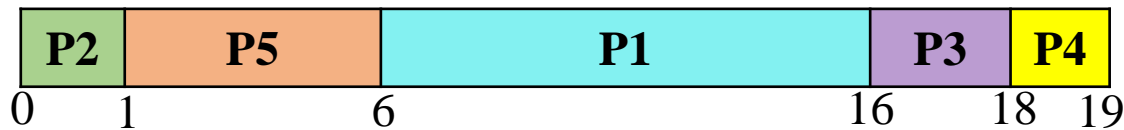
$$\text{الویت} = \frac{\text{زمان سرویس} + \text{زمان انتظار}}{\text{زمان سرویس}}$$

زمانبندی الویت Priority Scheduling

□ این زمانبندی بر اساس الویتی است که همراه مشخصات پردازش تعیین می گردد. هم به صورت انحصاری و هم غیر انحصاری است. در غیر انحصاری کار با الویت بالاتر اگر وارد صف شد، جای کار در حال پردازش را می گیرد، و در انحصاری، در ابتدای صف قرار می گیرد.

میانگین زمان انتظار را برای پردازشهای زیر در الگوریتم الویت را محاسبه کنید. (عدد کوچکتر یعنی الویت بیشتر)

نام پردازش	P1	P2	P3	P4	P5
زمان انفجار CPU	10	1	2	1	5
الویت	3	1	3	4	2



نمودار گانت الویت

$$\text{میانگین زمان انتظار} = \frac{0 + 1 + 6 + 16 + 18}{5} = 8.2$$

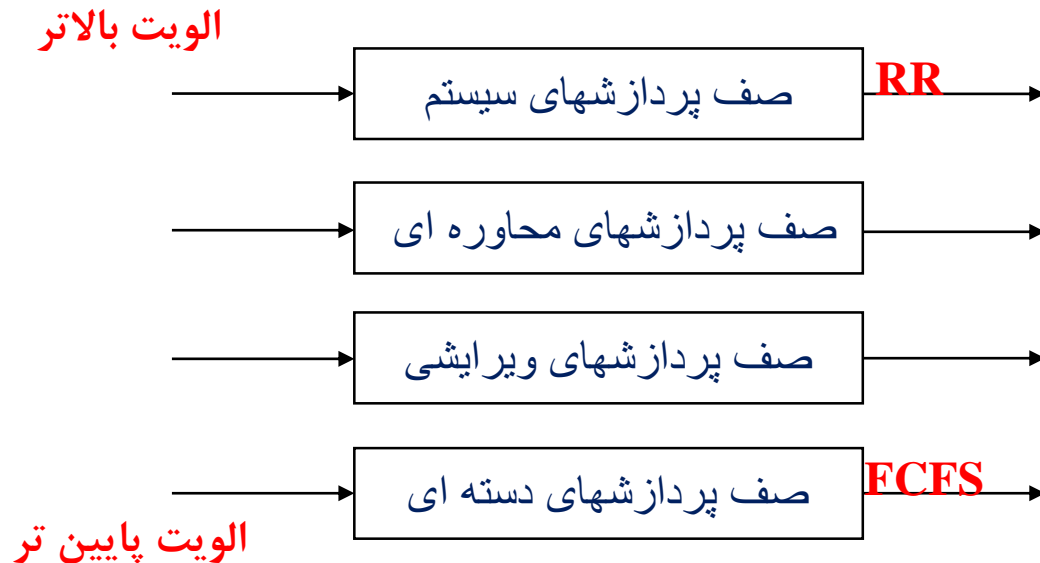
پیری Aging

□ در صورتیکه کاری با الویت کم باشد ممکن است مدت زیادی منتظر بماند، به آن پدیده پیری (aging) گفته می شود، یعنی دچار قحطی زدگی می شود. بنابراین هر مدت زمان مشخصی که پردازش منتظر بماند، یک واحد الویتش بیشتر می شود.

زمانبندی صفهای چند گانه Multiple Queue

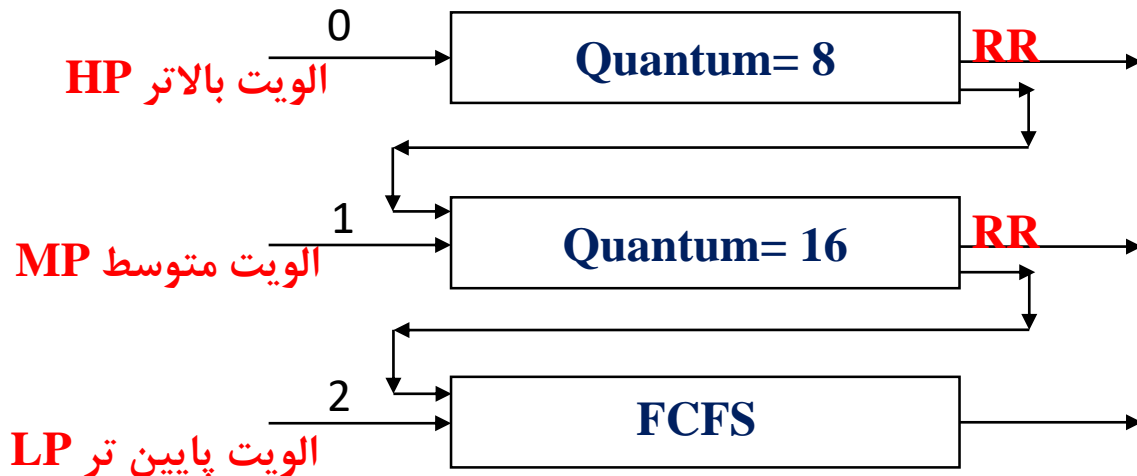
□ در این روش صفهایی در قسمت آماده درست می شود، که پردازشهای محاوره ای (foreground) و پردازشها پس زمینه (background) از هم جدا می شوند. هر صف الویتی مشخص دارد و درون صفها نیز الگوریتمهای زمانبندی متفاوتی ممکن است داشته باشند. ابتدا پردازشهای درون صف با الویت بالاتر اجرا می شوند. از این نظر زمانبندی انحصاری می باشد.

□ به عنوان مثال سیستمی ممکن است دارای صفهای زیر باشد:



زمانبندی صفهای چند گانه با فیدبک MFQ Multiple Feedback Queue

- در این روش صفهای با الویت بیشتر دارای کوانتوم کمتر می باشند. غیر انحصاری و روش RR در صفها استفاده می شود. اگر پردازشی در کوانتوم مورد نظر کارش تمام نشد به انتهای صف با الویت کمتر می رود.
- به عنوان مثال سیستمی ممکن است دارای صفهای زیر باشد:



□ پارامترهای این الگوریتم:

- ۱- تعداد صفها.
- ۲- الگوریتم زمانبندی هر صف.
- ۳- چه زمان پردازش به صف با لاویت متفاوت برود.
- ۴- پردازش هنگام ورود به کدام صف برود.

زمانبندی شانس **Lottery Scheduling**

□ در این روش سیستم عامل به هر پردازش تعدادی عدد اختصاص می دهد، هر چه الویت پردازش بیشتر باشد، تعداد این اعداد بیشتر می شود.

سپس یک عدد تولید می شود و هر پردازشی که این عدد را دارد، CPU به آن تعلق می گیرد.

زمانبندی **Longest Processing Time LPT**

□ در سیستمهای چند پردازنده ای، هرگاه یک پردازنده آزاد می گردد به پردازش با طولانی ترین زمان اجرا داده می شود. معمولاً انحصاری می باشد.