دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

سیستم‌های عامل

(فصل سوم : زمان‌بندی پردازه)

مهدی دادبخش

mahdi.dadbakhsh@sharif.edu

شماره درس : ۴۰۴۲۴ – گروه ۲

شنبه – دوشنبه (۱۶:۳۰ الی ۱۸:۰۰)

۱۴۰۲ – ۱۴۰۱

زمان‌بندی

دسته‌بندی الگوریتم‌های زمان‌بندی

معیارهای انتخاب یک الگوریتم

مفاهیم و اختصارات

معرفی الگوریتم‌ها

زمان‌بندی نخ

زمان‌بندی چند پردازنده‌ای

پایان

زمان‌بندی (Scheduling)

- هدف از زمان‌بندی، تخصیص پردازنده به پردازه‌ها می‌باشد به‌طوری‌که مواردی نظیر زمان پاسخ، توان عملیاتی و کارایی پردازنده در نظر گرفته شود.

- به‌طور کلی زمان‌بندی به سه نوع مختلف مطرح می‌شود :

❖ زمان‌بندی طولانی مدت (Long Term Scheduling) :

زمان‌بندی طولانی مدت یعنی انتخاب یکی از برنامه‌های موجود در حافظه جانبی و انتقال آن به حافظه اصلی. این زمان‌بندی درجه چندبرنامگی را کنترل می‌کند. (تغییر حالت از new به ready یا تغییر حالت از new به suspend ready)

❖ زمان‌بندی میان مدت (Medium Term Scheduling) :

در این مرحله، تعیین می‌شود که چه پردازه یا پردازه‌هایی از حالت معلق خارج شده و به حافظه اصلی برگردانده شوند و برعکس.

(تغییر حالت‌های suspend ready به ready ، suspend wait به wait ، suspend ready به ready و wait به suspend wait)

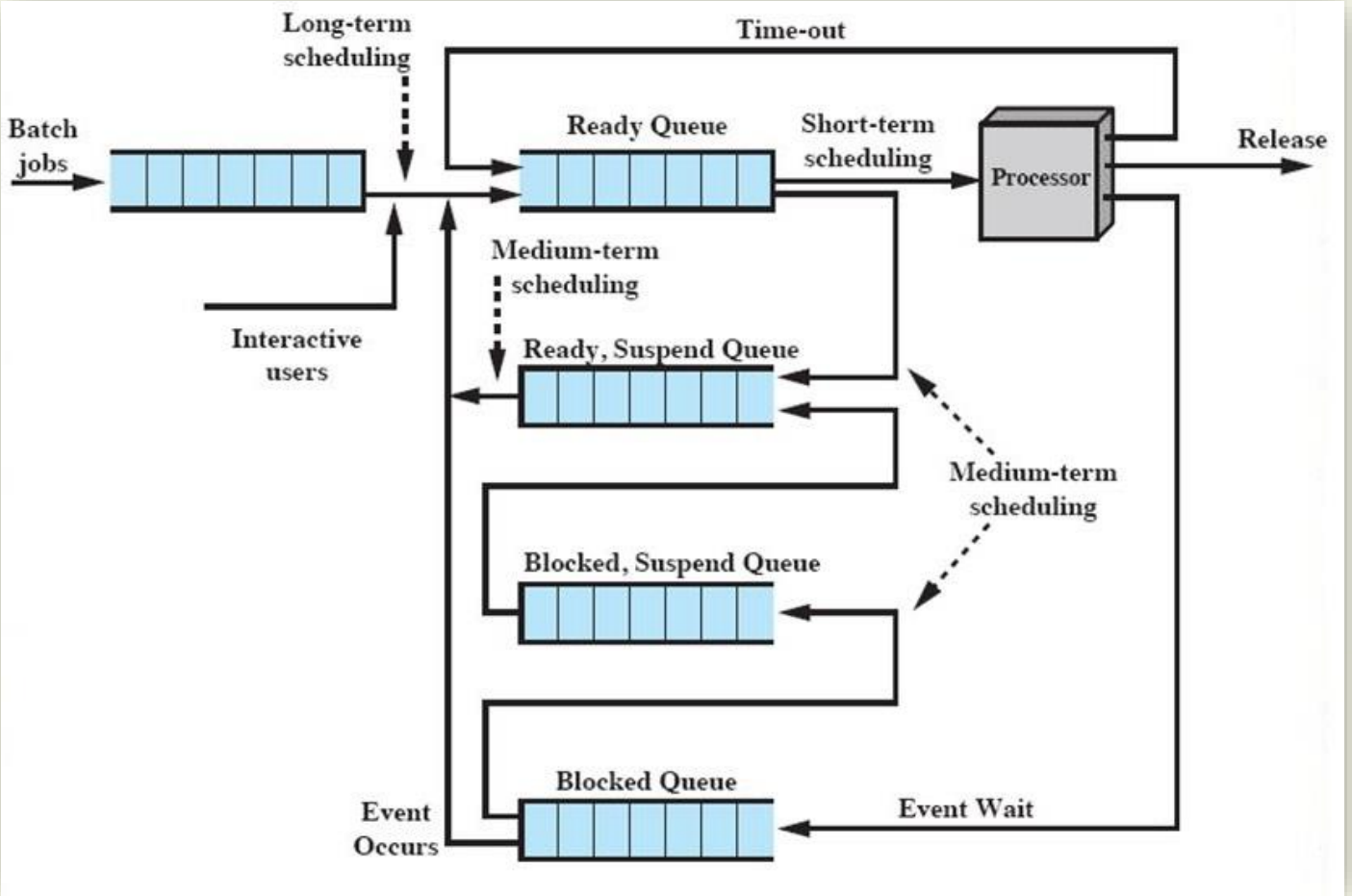
❖ زمان‌بندی کوتاه مدت (Short Term Scheduling) :

در این مرحله، تعیین می‌شود که چه پردازه ای باید پردازنده را در اختیار بگیرد (تغییر حالت از ready به running).

در واقع تصمیم گیری در مورد انتخاب یکی از پردازه‌های موجود در صف آماده و تخصیص پردازنده به آن ، در این مرحله انجام می‌شود.

نمودار صف برای زمان‌بندی

نمودار صف برای زمان‌بندی



دسته‌بندی الگوریتم‌های زمان‌بندی

- منظور از الگوریتم‌های زمان‌بندی، زمان‌بندی کوتاه مدت است. هدف اصلی زمان‌بندی کوتاه مدت، تعیین و انتخاب یکی از پردازه‌های موجود در صف Ready می‌باشد. برای این منظور الگوریتم‌های متعددی وجود دارد. این الگوریتم‌ها از نظر نحوه‌ی تخصیص پردازنده در دو دسته کلی قرار می‌گیرند :

- دسته اول : الگوریتم‌های انحصاری :

  در این روش‌ها، وقتی پردازه‌ای در حا اجرا باشد و پردازنده را در اختیار داشته باشد، سیستم عامل نمی‌تواند پردازنده را از او بگیرد، مگر اینکه کارش به پایان برسد (Terminate) و یا اینکه به عملیات ورودی/خروجی نیاز داشته باشد (Wait) . این بدان معنی است که پردازه تا زمانی که به پردازنده نیاز دارد می‌تواند آن را در اختیار داشته باشد.

- دسته دوم : الگوریتم‌های غیرانحصاری :

  در این روش‌ها، می‌توان بنا به دلایل خاصی پردازنده را از پردازه در حال اجرا گرفت و به پردازه‌ی دیگری تخصیص داد. این دلایل می‌تواند یکی از موارد زیر باشد :

  الف) به پایان رسیدن کار پردازه

  ب) نیاز به عملیات ورودی/خروجی و یا رخ دادن وقفه‌ای دیگر

  ج) به پایان رسیدن بازه زمانی که پردازنده در اختیار پردازه بوده است.

  د) ورود پردازه‌ای با اولویت بیشتر.

معیارهای انتخاب یک الگوریتم

- زمان برگشت (Turnaround Time) :
  این زمان مدت زمانی است که صرف می‌شود تا یک پردازه به پایان برسد. به عبارت دیگر، فاصله زمانی بین زمان پذیرش یک پردازه تا زمان تکمیل آن.
  این زمان شامل زمان اجرا و زمان صرف شده جهت انتظار برای منابع (منابع غیرپردازنده و پردازنده) می‌باشد.
  این زمان معیار مناسبی برای پردازه‌های دسته‌ای می‌باشد.

- زمان پاسخ (Response Time) :
  این زمان مدت زمان صرف شده بین ارائه یک تقاضا تا دریافت پاسخ می‌باشد. به عبارت دیگر، این زمان مدت زمانی است که صرف پردازش (اجرا) و انتظار برای پردازنده می‌شود و شامل انتظار برای سایر منابع نیست یعنی مواردی را که پردازه به حالت wait می‌رود شامل نمی‌شود.

- توان عملیاتی (Throughput) :
  عبارت است از تعداد پردازه به پایان رسیده در واحد زمان. این معیار بیان کننده مقدار کارا نجام شده در واحد زمان می‌باشد.

- استفاده از پردازنده (Processor Utilization) :
  بیاتگر درصد مشغول بودن پردازنده است.

- عدالت (Fairness) :
  منظور از عدالت این است که پردازه‌ها به میزان یکسانی از وقت پردازنده اتسفاده کنند و هیچ یک از پردازه‌ها دچار گرسنگی (Starvation) نشود.

تحقیق ۵ : مفهوم گرسنگی به چه معنایی می‌باشد؟(۱ نفر)

مفاهیم و اختصارات

مفهوم	نام اختصاری	شرح
زمان رسیدن یا ورود (Arrive Time)	AT	زمان ورود پردازه به سیستم
زمان پایان (Final Time)	FT	زمان پایان کار پردازه و خروج آن از سیستم
زمان پاسخدهی (Response Time)	RT	مدت زمانی که صرف می‌شود تا پردازه به نتیجه برسد، یعنی مجموع زمانی که پردازه در حال اجرا است و زمانی که در صف آماده قرار دارد.
زمان انتظار (Waiting Time)	WT	مدت زمانی است که پردازه در سیستم وجود دارد ولی پردازنده را در اختیار ندارد.
متوسط زمان پاسخدهی (Average Response Time)	ART	میانگین زمام پاسخدهی تمام پردازه‌های موجود در سیستم
متوسط زمان انتظار (Average Waiting Time)	AWT	میانگین زمان انتظار تمام پردازه‌های موجود در سیستم
زمان پردازش (CPU Burst Time)	CBT	مدت زمانی است که پردازه به پردازنده نیاز دارد.
زمان تعویض متن (Context Switch)	CS	مدت زمانی است که صرف می‌شود تا یک پردازه از حالت اجرا خارج شده و پردازه دیگر جایگزین آن گردد.
برش زمانی (Time Slice)	TS	بازه زمانی است پردازنده به پردازه‌ها اختصاص داده می‌شود.
تعداد پردازه	n	تعداد پردازه‌های موجود در سیستم

$$RT = FT - AT \;,\; WT = RT - CBT \;,\; ART = \frac{\sum_{i=1}^{n} RT[i]}{n} \;,\;,\; AWT = \frac{\sum_{i=1}^{n} WT[i]}{n}$$

معرفی الگوریتم‌ها

First In First Out	Shortest Job First	Longest Process Time
Highest Response Radio Next	Round Robin	Shortest Remaining Time First
Priority	Three Level Scheduling	Multi Level Queue
Multi Level Feedback Queue	Guaranteed	Lottery
Fair Share Scheduling	Real Time Scheduling	Deadline Scheduling
مقایسه برخی از الگوریتم‌ها	زمان تعویض متن غیر صفر	مثال

مثال یک

فرض کنید پنج پردازه در یک سیستم اشتراک زمانی وجود دارد. اطلاعات مربوط به زمان ورود، زمان پردازش و اولویت پردازه‌ها در جدول زیر آمده است. می‌خواهیم متوسط زمان پاسخدهی و متوسط زمان انتظار پردازه‌ها را در هر یک از الگوریتم‌ها بدست آوریم :

Process	AT	CBT	Priority
P0	0	3	3
P1	2	6	2
P2	4	4	1
P3	6	5	3
P4	8	2	2

نکته ۱ : عدد کمتر در ستون priority به معنای اولویت بیشتر است.

نکته ۲ : برای حل این مثال، از نمودار میله‌ای (گانت) استفاده می‌کنیم. در بالای نمودار، زمان مشخص می‌شود و در زیر نمودار، پردازه‌های موجود درصف را در زمان‌های مختلف نشان می‌دهیم. همچنین در داخل نمودار پردازه‌های در حال اجرا نمایش داده می‌شوند.

الگوریتم اول : خروج به ترتیب ورود (FIFO - First In First Out)

- این روش، یک روش انحصاری است. در این روش، از بین پردازه‌های موجود در صف ready ، پردازه‌ای را انتخاب می‌کنیم که زودتر وارد صف شده باشد و به عبارت دیگر، زمان ورودی کمتری داشته باشد.

- **نکته ۱ (خیلی مهم) :** در تمامی روش‌ها و الگوریتم‌ها، ابتدا بر حسب زمان ورود، صف ready را تشکیل می‌دهیم. سپس بر حسب معیار الگوریتم مربوطه از صف انتخاب می‌کنیم.

- **نکته ۲ :** در نمودار گانت، همیشه زمان را از صفر شروع می‌کنیم حتی اگر پردازه‌ای در زمان صفر وارد نشده باشد. در این صورت، از زمان صفر تا زمان ورود اولین پردازه نمودار را هاشور می‌زنیم. به همین ترتیب، اگر در هر بازه‌ای از زمان پردازه‌ای در صف وجود نداشته باشد در آن بازه نمودار هاشور خورده و پردازنده بیکار خواهد بود.

- **نکته ۳ :** الگوریتم FIFO برای پردازه‌های طولانی مدت بسیار بهتر از پردازه‌های کوتاه مدت عمل می‌کند .

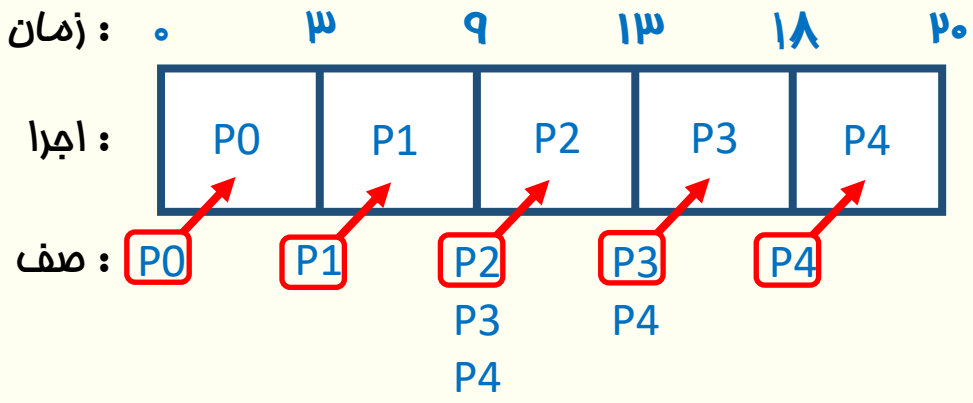
- **سوال ۱ (خیلی مهم) :** اگر در زمان تشکیل صف، زمان ورود دو یا چند پردازه برابر باشد، چه باید کرد و کدام پردازه انتخاب می‌شود؟

- **سوال ۲ :** در مواقعی که پردازه‌های I/O Bound و CPU Bound در یک صف قرار داشته باشند، الگوریتم FIFO روش مناسبی نیست. چرا ؟

حل مثال یک به روش FIFO

مثال حل شده به روش FIFO

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۳	۰
P1	۲	۶		۷	۱
P2	۴	۴		۹	۵
P3	۶	۵		۱۲	۷
P4	۸	۲		۱۲	۱۰



ART = (۳ + ۷ + ۹ + ۱۲ + ۱۲) / ۵ = ۴۲ / ۵ = ۸.۶

AWT = (۰ + ۱ + ۵ + ۷ + ۱۰) / ۵ = ۲۳ / ۵ = ۴.۶

نمونه سوال

نمونه سوال برای روش FIFO

فرض کنید شش پردازه مطابق جدول زیر وارد سیستم می‌شوند. مطلوب است :

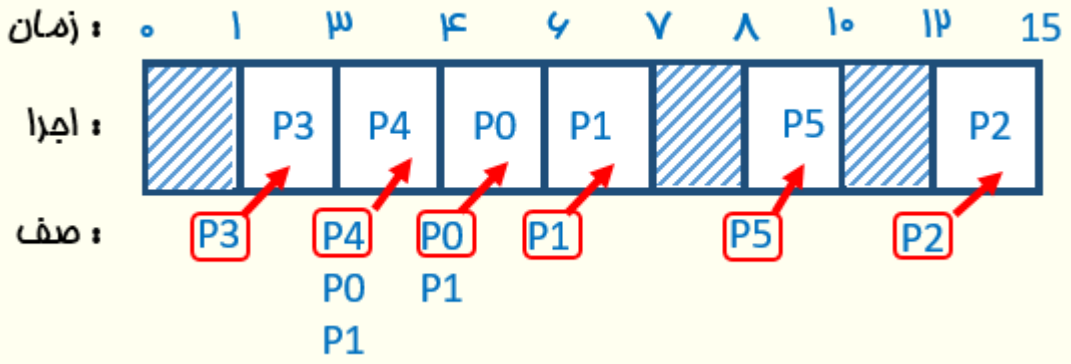
- الف) متوسط زمان پاسخدهی
- ب) متوسط زمان انتظار
- ج) مدت زمان بیکاری پردازنده
- د) درصد کارایی پردازنده

Process	AT	CBT
P0	۲	۲
P1	۳	۱
P2	۱۲	۳
P3	۱	۲
P4	۲	۱
P5	۸	۲

جواب

جواب سوال

Process	AT	CBT	FT	RT	WT
P0	۲	۲	۶	۴	۲
P1	۳	۱	۷	۴	۳
P2	۱۲	۳	۱۵	۳	۰
P3	۱	۲	۳	۲	۰
P4	۲	۱	۴	۲	۱
P5	۸	۲	۱۰	۲	۰



ART = (۴ + ۴ + ۳ + ۲ + ۲ + ۲) / ۶ = ۱۷ / ۶ = ۲.۸۳

AWT = (۲ +۳ + ۰ + ۰ + ۱ + ۰) / ۶ = ۶ / ۶ = ۱

CPU idle Time = ۴

CPU performance = (۱۵ – ۴) / ۱۵ * ۱۰۰ = ۷۳.۳۳ %

الگوریتم دوم : کوتاه‌ترین کار (SJF - Shortest Job First)

- این روش نیز یک روش انحصاری است. در این روش، از بین پردازه‌های موجود در صف ready ، پردازه‌ای را انتخاب می‌کنیم که زمان پردازش کمتری داشته باشد. مشکل روش SJF این است که باید زمان پردازش پردازه‌ها را بدانیم.

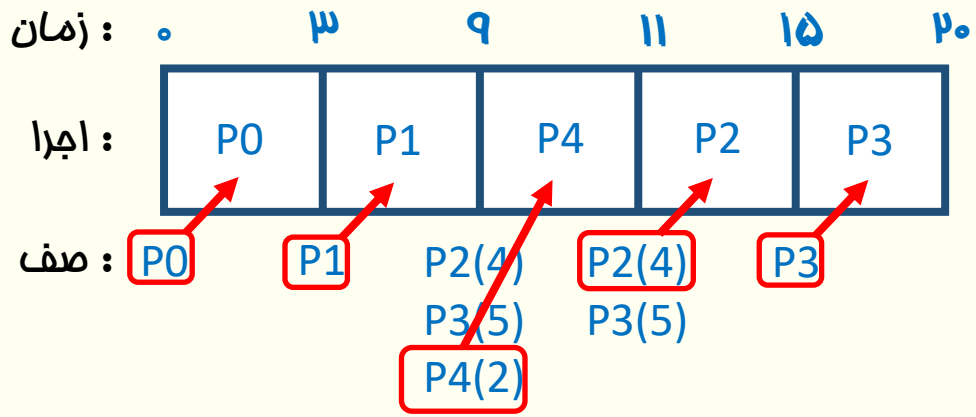
- **نکته ۱ : اگر زمان پردازش دو یا چند پردازه‌ی موجود در صف با هم برابر باشد پردازه ای انتخاب می‌شود که زودتر وارد صف شده باشد**

- **نکته ۲ : اگر زمان پردازش همه پردازه‌ها با هم برابر باشد عملکرد روش همانند FIFO خواهد بود .**

حل مثال یک

مثال حل شده به روش SJF

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۳	۰
P1	۲	۶		۷	۱
P2	۴	۴		۱۱	۷
P3	۶	۵		۱۴	۹
P4	۸	۲		۳	۱



AWT = (۰ + ۱ + ۷ + ۹ + ۱) / ۵ = ۱۸ / ۵ = ۳.۶

ART = (۳ + ۷ + ۱۱ + ۱۴ + ۳) / ۵ = ۳۸ / ۵ = ۷.۶

نمونه سوال یک

نمونه سوال دو

نمونه سوال اول برای روش SJF

فرض کنید شش پردازه مطابق جدول زیر وارد سیستم می‌شوند. مطلوب است :

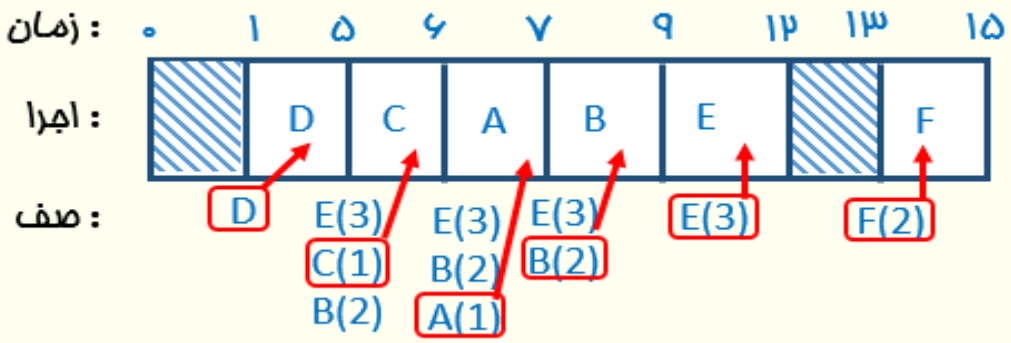
- الف) متوسط زمان پاسخدهی
- ب) متوسط زمان انتظار
- ج) مدت زمان بیکاری پردازنده
- د) درصد کارایی پردازنده

Process	AT	CBT
A	۶	۱
B	۳	۲
C	۳	۱
D	۱	۴
E	۲	۳
F	۱۳	۲

جواب

جواب سوال یک

Process	AT	CBT	FT	RT	WT
A	۶	۱	۷	۱	۰
B	۳	۲	۹	۶	۴
C	۳	۱	۶	۳	۲
D	۱	۴	۵	۴	۰
E	۲	۳	۱۲	۱۰	۷
F	۱۳	۲	۱۵	۲	۰



ART = (۱ + ۶ + ۳ + ۴ + ۱۰ + ۲) / ۶ = ۲۶ / ۶ = ۴.۳۳

AWT = (۰ +۴ + ۲ + ۰ + ۷ + ۰) / ۶ = ۱۳ / ۶ = ۲.۱۷

CPU idle Time = ۲

CPU performance = (۱۵ – ۲) / ۱۵ * ۱۰۰ = ۸۶.۶۷ %

نمونه سوال دوم برای روش SJF

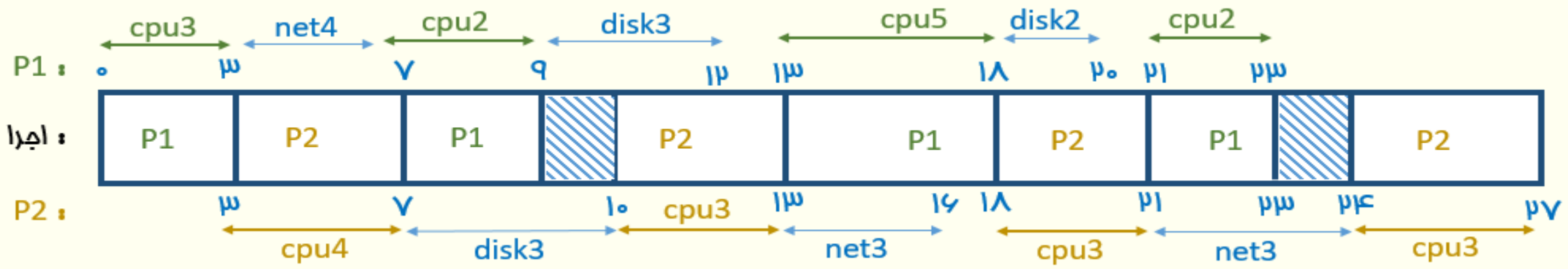
دو پردازه P1 و P2 با مشخصات اجرای زیر در سیستم موجود هستند. اطلاعات هر سطر منبع مورد نیاز برای هر پردازه و زمان مورد نیاز را مشخص می‌کند. مثلاً net3 در سطر چهارم بیانگر این است که پردازه دوم، کارت شبکه را به مدت ۳ ثانیه نیاز دارد. اگر پردازه P2 ۲ ثانیه بعد از P1 وارد سیستم شده باشد و سیستم سیاست SJF را برای زمان‌بندی پردازه‌ها اعمال نماید، زمان کل اجرای دو پردازه و همچنین زمان هدررفتگی وقت cpu را بر حسب ثانیه محاسبه کنید؟

- الف) کل زمان ۲۴ و هدررفتگی صفر
- ب) کل زمان ۲۵ و هدررفتگی ۱
- ج) کل زمان ۲۷ و هدررفتگی ۲
- د) کل زمان ۲۸ و هدررفتگی ۳

P1	P2
cpu3	cpu4
net4	disk3
cpu2	cpu3
disk3	net3
cpu5	cpu3
disk2	net3
Cpu2	cpu3

جواب

جواب سوال دو



Total Time = ۲۷

Idle Time = ۲

الگوریتم سوم : بیشترین زمان پردازش (LPT - Longest Process Time)

- این روش نیز یک روش انحصاری است. در این روش، از بین پردازه‌های موجود در صف ready ، پردازه‌ای را انتخاب می‌کنیم که زمان پردازش بیشتری داشته باشد. در اینجا نیز همانند روش SJF باید زمان پردازش پردازه‌ها را بدانیم.

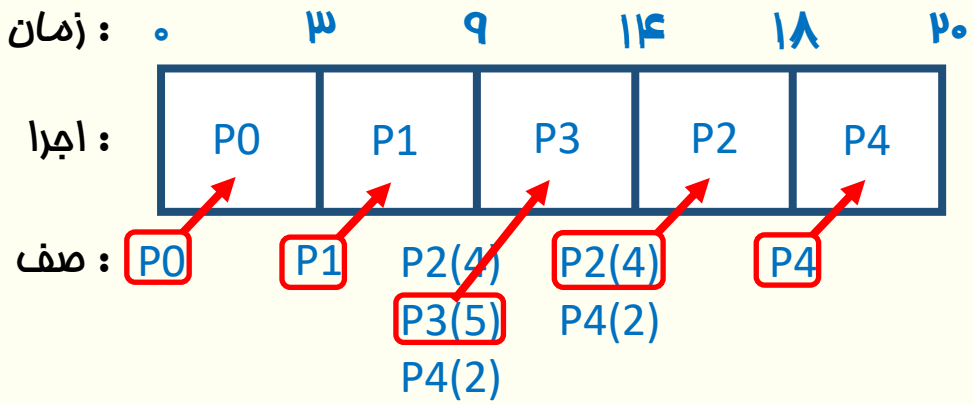
- **نکته ۱ : اگر زمان پردازش دو یا چند پردازه‌ی موجود در صف با هم برابر باشد پردازه ای انتخاب می‌شود که زودتر وارد صف شده باشد**

- **نکته ۲ : اگر زمان پردازش همه پردازه‌ها با هم برابر باشد عملکرد روش همانند FIFO خواهد بود .**

حل مثال یک

مثال حل شده به روش LPT

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۳	۰
P1	۲	۶		۷	۱
P2	۴	۴		۱۴	۱۰
P3	۶	۵		۸	۳
P4	۸	۲		۱۲	۱۰



ART = (۳ + ۷ + ۱۴ + ۸ + ۱۲) / ۵ = ۴۴ / ۵ = ۸.۸

AWT = (۰ + ۱ + ۱۰ + ۳ + ۱۰) / ۵ = ۲۴ / ۵ = ۴.۸

نمونه سوال

نمونه سوال برای روش LPT

فرض کنید شش پردازه مطابق جدول زیر وارد سیستم می‌شوند. مطلوب است :

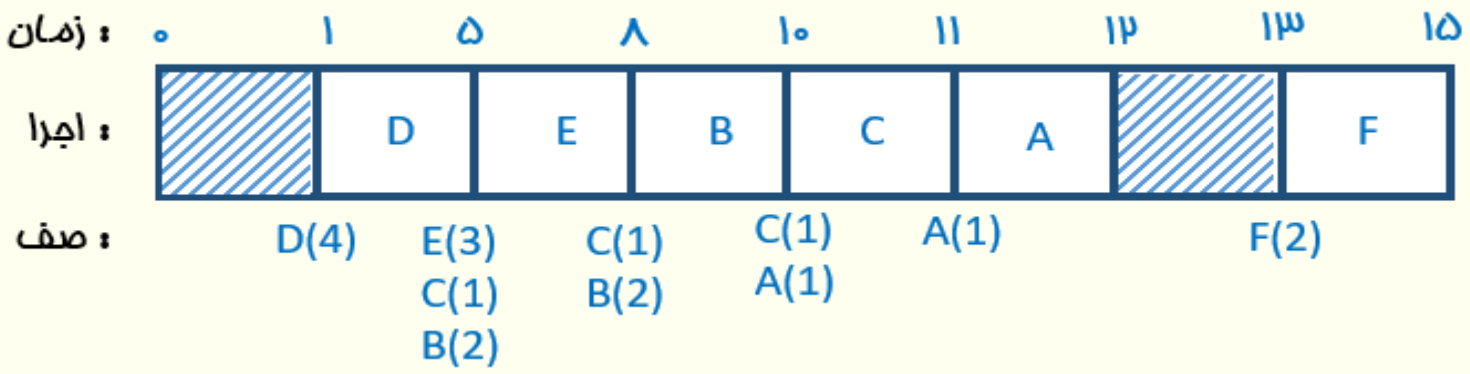
- الف) متوسط زمان پاسخدهی
- ب) متوسط زمان انتظار
- ج) مدت زمان بیکاری پردازنده
- د) درصد کارایی پردازنده

Process	AT	CBT
A	۶	۱
B	۳	۲
C	۳	۱
D	۱	۴
E	۲	۳
F	۱۳	۲

جواب

جواب سوال

Process	AT	CBT	FT	RT	WT
A	۶	۱	۱۲	۶	۵
B	۳	۲	۱۰	۷	۵
C	۳	۱	۱۱	۸	۷
D	۱	۴	۵	۴	۰
E	۲	۳	۸	۶	۳
F	۱۳	۲	۱۵	۲	۰



ART = (۶ + ۷ + ۸ + ۴ + ۶ + ۲) / ۶ = ۳۳ / ۶ = ۵.۵

AWT = (۵ + ۵ + ۷ + ۰ + ۳ + ۰) / ۶ = ۲۰ / ۶ = ۳.۳۳

CPU idle Time = ۲

CPU performance = (۱۵ – ۲) / ۱۵ * ۱۰۰ = ۸۶.۶۷ %

الگوریتم چهارم : بالاترین نرخ پاسخ (HRRN - Highest Response Ration Next)

- این روش نیز یک روش انحصاری است. در این روش، از بین پردازه‌های موجود در صف ready ، پردازه‌ای را انتخاب می‌کنیم که نرخ پاسخدهی بیشتری داشته باشد. معیار نرخ پاسخدهی، در کتاب‌های مختلف، متفاوت است . نرخ پاسخدهی را به صورت زیر بدست می‌آوریم :

$$\text{نرخ پاسخدهی} = \frac{\text{میزان انتظار}}{\text{زمان پردازش}} = \frac{\text{زمان جاری} - \text{زمان ورود}}{\text{زمان پردازش}}$$

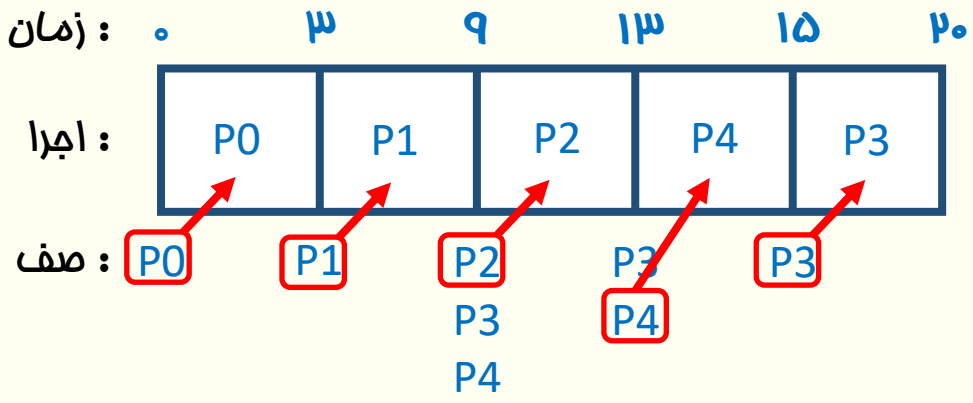
$$\text{Response Ratio} = \frac{Wait}{CBT} = \frac{t\ (current\ time\ ) - AT}{CBT}$$

- نکته ۱ : اگر نرخ پاسخدهی دو یا چند پردازه موجود در صف با هم برابر باشد، پردازه‌ای انتخاب می‌شود که زمان پردازش کمتری داشته باشد.
- نکته ۲ : اگر زمان ورود تمام پردازه‌ها برابر باشد، عملکرد روش همانند روش SJF خواهد بود.
- نکته ۳ : اگر زمان پردازش تمام پردازه‌ها برابر باشد، عملکرد روش همانند روش FIFO خواهد بود.

حل مثال یک

مثال حل شده به روش HRRN

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۳	۰
P1	۲	۶		۷	۱
P2	۴	۴		۹	۵
P3	۶	۵		۱۴	۹
P4	۸	۲		۷	۵



$$T = 9 : \begin{cases} Ratio\ (P2) = \frac{(9-4)}{4} = \frac{5}{4} \\ Ratio\ (P3) = \frac{(9-6)}{5} = \frac{3}{5} \\ Ratio\ (P4) = \frac{(9-8)}{2} = \frac{1}{2} \end{cases}$$

$$T = 13 : \begin{cases} Ratio\ (P3) = \frac{(13-6)}{5} = \frac{7}{5} \\ Ratio\ (P4) = \frac{(13-8)}{2} = \frac{5}{2} \end{cases}$$

ART = (۳ + ۷ + ۹ + ۱۴ + ۷) / ۵ = ۴۰ / ۵ = ۸

AWT = (۰ + ۱ + ۵ + ۹ + ۵) / ۵ = ۲۰ / ۵ = ۴

نمونه سوال

نمونه سوال برای روش HRRN

فرض کنید شش پردازه مطابق جدول زیر وارد سیستم می‌شوند. مطلوب است :

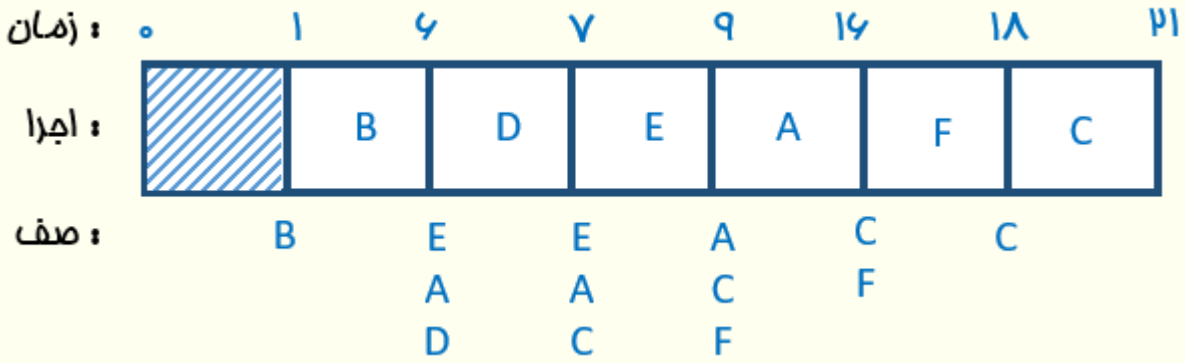
- الف) متوسط زمان پاسخدهی
- ب) متوسط زمان انتظار
- ج) مدت زمان بیکاری پردازنده
- د) درصد کارایی پردازنده

Process	AT	CBT
A	۴	۷
B	۱	۵
C	۷	۳
D	۴	۱
E	۲	۲
F	۸	۲

جواب سوال

جواب سوال

Process	AT	CBT	FT	RT	WT
A	۴	۷	۱۶	۱۲	۵
B	۱	۵	۶	۵	۰
C	۷	۳	۲۱	۱۴	۱۱
D	۴	۱	۷	۳	۲
E	۲	۲	۹	۷	۵
F	۸	۲	۱۸	۱۰	۸



$$T = 6 : \begin{cases} E = \frac{(6-2)}{2} = 2 \\ A = \frac{(6-4)}{7} = \frac{2}{7} \\ D = \frac{(6-4)}{1} = 2 \end{cases}$$

$$T = 7 : \begin{cases} E = \frac{(7-2)}{2} = \frac{5}{2} \\ A = \frac{(7-4)}{7} = \frac{3}{7} \\ C = \frac{(7-7)}{3} = 0 \end{cases}$$

$$T = 9 : \begin{cases} A = \frac{(9-4)}{7} = \frac{5}{7} \\ C = \frac{(9-7)}{3} = \frac{2}{3} \\ F = \frac{(9-8)}{2} = \frac{1}{2} \end{cases}$$

$$T = 16 : \begin{cases} C = \frac{(16-7)}{3} = \frac{9}{3} \\ F = \frac{(16-8)}{2} = \frac{8}{2} \end{cases}$$

ART = (۱۲ + ۵ + ۱۴ + ۳ +۷ + ۱۰) / ۶ = ۵۱ / ۶ = ۸.۵

AWT = (۵ + ۰ + ۱۱ + ۲ + ۵ + ۸) / ۶ = ۳۱ / ۶ = ۵.۱۶۷

CPU idle Time = ۱

CPU performance = (۲۱ – ۱) / ۲۱ * ۱۰۰ = ۹۵.۲۴ %

الگوریتم پنجم : نوبت گردشی (RR - Round Robin)

- این روش، برخلاف روش‌های قبل یک روش غیرانحصاری است. در این روش، همانند روش FIFO از بین پردازه‌های موجود در صف ready ، پردازه‌ای را انتخاب می‌کنیم که زودتر وارد صف شده باشد، با این تفاوت که به هر پردازه به اندازه یک بازه زمانی معین (TS) پردازنده را اختصاص می‌دهیم. اگر کار پردازه در این بازه به اتمام برسد (یعنی زمان پردازش پردازه کمتر یا مساوی برش زمانی مورد نظر باشد)، پردازه از سیستم خارج می‌شود، ولی اگر نیاز پردازه بیش از برش زمانی باشد و کارش تمام نشود، باید به انتهای صف ready برود و منتظر بماند تا دوباره نوبت به او برسد. این کار ممکن است بر حسب میزان نیاز پردازه به دفعات تکرار شود.

- نکته ۱ : اگر برش زمانی خیلی کوچک باشد، به دلیل وجود زمان تعویض متن بین پردازه‌ها عملکرد سیستم کاهش می‌یابد.

- نکته ۲ : اگر برش زمانی خیلی بزرگ باشد (از بیشترین زمان پردازش نیز بزرگ‌تر باشد) ، عملکرد روش همانند روش FIFO خواهد بود.

- بر حسب میزان برش زمانی، مسائل مربوط به این روش به دو دسته تقسیم می‌شوند و برای هر یک راه حل متفاوتی وجود دارد :

  - ❖ حالت اول : برش زمانی تقریبا نزدیک به زمان‌های پردازش باشد.
  - ❖ حالت دوم : برش زمانی خیلی کوچک‌تر از زمان‌های پردازش باشد.

حالت اول

$TS \simeq CBT$

حالت دوم

$TS << CBT$

حالت اول روش Round Robin

- در این حالت، برش زمانی و زمان‌های پردازش از نظر واحد زمانی یکسان هستند و از نظر مقدار، نزدیک به هم می‌باشند. فرض کنید در مثال مطرح شده، زمانهای پردازش برحسب میلی ثانیه و برش زمانی برابر با دو میلی ثانیه باشد.
- برای حل مسائل این حالت همانند قبل از نمودار گانت استفاده می‌کنیم.
- نکته : تشکیل صف در این روش (به طور کلی در روش‌های غیر انحصاری) با روش‌های قبلی (روش‌های انحصاری) متفاوت است.
- در این روش، صف به صورت زیر تشکیل می‌شود :
  - ❖ ابتدا پردازه‌هایی در صف قرار می‌گیرند که از قبل در صف بوده‌اند (به جز پردازه در حال اجرا).
  - ❖ سپس پردازه‌هایی در صف قرار می‌گیرند که در بازه زمانی آخر وارد سیستم شده باشند.
  - ❖ در نهایت، پردازه در حال اجرا اگر کارش به اتمام نرسیده باشد، به انتهای صف اضافه می‌شود.

حل مثال یک

مثال حل شده برای حالت اول روش RR

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۵	۲
P1	۲	۶		۱۵	۹
P2	۴	۴		۹	۵
P3	۶	۵		۱۴	۹
P4	۸	۲		۷	۵

ART = (۵ + ۱۵ + ۹ + ۱۴ + ۷) / ۵ = ۵۰ / ۵ = ۱۰

AWT = (۲ + ۹ + ۵ + ۹ + ۵) / ۵ = ۳۰ / ۵ = ۶

زمان :	۰	۲	۴	۵	۷	۹	۱۱	۱۳	۱۵	۱۷	۱۹	20
اجرا :	P0	P1	P0	P2	P1	P3	P2	P4	P1	P3	P3	
صف :	P0(3)	P1(6) P0(1)	P0(1) P2(4) P1(4)	P2(4) P1(4)	P1(4) P3(5) P2(2)	P3(5) P2(2) P4(2) P1(2)	P2(2) P4(2) P1(2) P3(3)	P4(2) P1(2) P3(3)	P1(2) P3(3)	P3(3)	P3(1)	

نمونه سوال

نمونه سوال برای حالت اول روش RR

فرض کنید پنج پردازه مطابق جدول زیر وارد سیستم می‌شوند. با فرض برش زمانی برابر ۵ ، مطلوب است :

- الف) متوسط زمان پاسخدهی
- ب) متوسط زمان انتظار

Process	AT	CBT
A	۱	۱۲
B	۲	۸
C	۳	۳
D	۴	۵
E	۵	۷

جواب سوال

جواب سوال مربوط به حالت اول

Process	AT	CBT	FT	RT	WT
A	۱	۱۲	۳۶	۳۵	۲۳
B	۲	۸	۳۲	۳۰	۲۲
C	۳	۳	۱۴	۱۱	۸
D	۴	۵	۱۹	۱۵	۱۰
E	۵	۷	۳۴	۲۹	۲۲

ART = (۳۵ + ۳۰ + ۱۱ + ۱۵ +۲۹) / ۵ = ۱۲۰ / ۵ = ۲۴

AWT = (۲۳ + ۲۲ + ۸ + ۱۰ + ۲۲) / ۶ = ۸۵ / ۵ = ۱۷

زمان : ۰ ۱ ۶ ۱۱ ۱۴ ۱۹ ۲۴ ۲۹ ۳۲ ۳۴ ۳۶

اجرا : A | B | C | D | E | A | B | E | A

صف :

A(12)	B(8)	C(3)	D(5)	E(7)	A(7)	B(3)	E(2)	A(2)
	C(3)	D(5)	E(7)	A(7)	B(3)	E(2)	A(2)	
	D(5)	E(7)	A(7)	B(3)	E(2)	A(2)		
	E(7)	A(7)	B(3)					
	A(7)	B(3)						

حالت دوم روش Round Robin

- مسائل مربوط به این حالت ممکن است به دو صورت مطرح شوند:
  - ❖ واحد زمانی برش زمانی حداقل یک واحد کمتر از واحد زمانی زمان‌های پردازش باشد. مثلاً برش زمانی پنج میلی ثانیه باشد و زمان‌های پردازه بر حسب ثانیه باشند.
  - ❖ واحد زمانی هر دو یکسان باشد ولی مقدار برش زمانی خیلی کمتر از زمان‌های پردازش باشد. مثلاً برش زمانی ۰/۰۱ ثانیه و زمان‌های پردازش نیز بر حسب ثانیه و بزرگتر از یک باشند.
- در این حالت نمی‌توان از نمودار گانت استفاده کرد، چرا که تعداد قسمت‌های نمودار بسیار زیاد خواهد شد. بنابراین در این حالت از نمودار دیگری که به صورت دستگاه مختصات است، استفاده می‌کنیم، به طوری که بر روی محور عمودی پردازه‌ها و بر روی محور افقی زمان را نشان می‌دهیم.
- در این حالت چون برش زمانی خیلی کوچک است در واقع زمان پردازنده بین پردازه‌های موجود به طور مساوی تقسیم می‌شود.
- زمان روی محور افقی در دو مرحله مشخص می‌شود :
  - ❖ در مرحله اول زمان‌های ورود پردازه‌ها را روی محور افقی درج می‌کنیم.
  - ❖ در مرحله دوم (پس از ورود آخرین پردازه) معیار تعیین زمان، پایان پردازه‌ها می‌باشد.
- نکته ۱ : در حالت دوم روش Round Robin زمان تعویض متن را صفر در نظر می‌گیریم.
- نکته ۲ : ابتدا در هر بازه زمانی سهم هر پردازه را مشخص می‌کنیم. اگر تمام پردازه‌ها بیشتر یا مساوی سهمشان نیاز داشتند، به‌طور عادی زمان آن بازه بین آنها به صورت مساوی تقسیم می‌شود.
- نکته ۳ : اگر حالتی پیش بیاید که پس از تعیین سهم هر پردازه، پردازه‌ای وجود داشته باشد که کمتر از سهمش نیاز داشته باشد، مسلماً زودتر از ورود پردازه بعدی کارش به پایان می‌رسد. در این حالت ابتدا باید بر اساس پایان این پردازه تصمیم بگیریم و سپس بر اساس ورود پردازه بعدی.

حل مثال یک

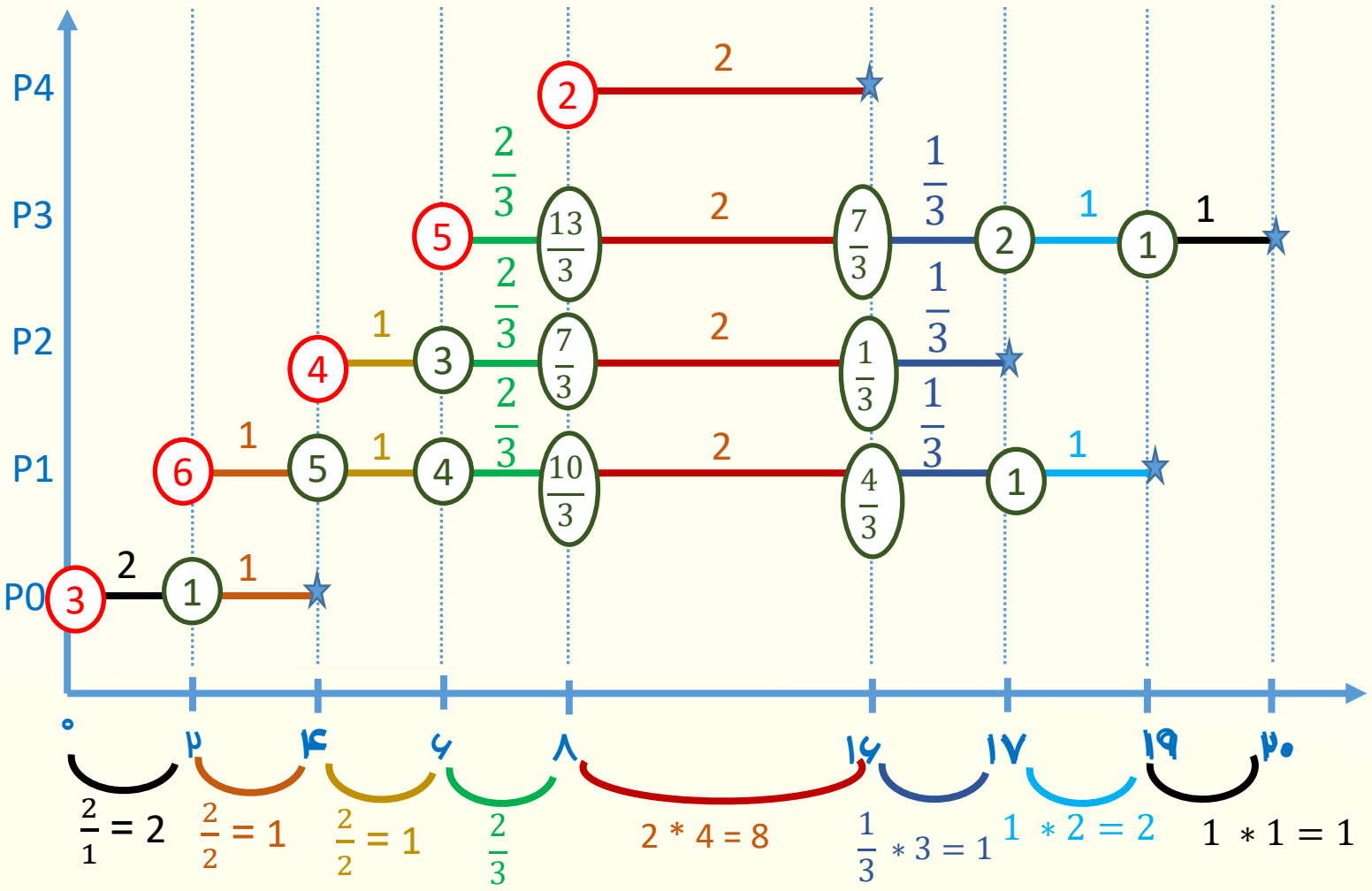
حل مثال برای نکته ۳

مثال حل شده برای حالت دوم روش RR

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۴	۱
P1	۲	۶		۱۷	۱۱
P2	۴	۴		۱۳	۹
P3	۶	۵		۱۴	۹
P4	۸	۲		۸	۶

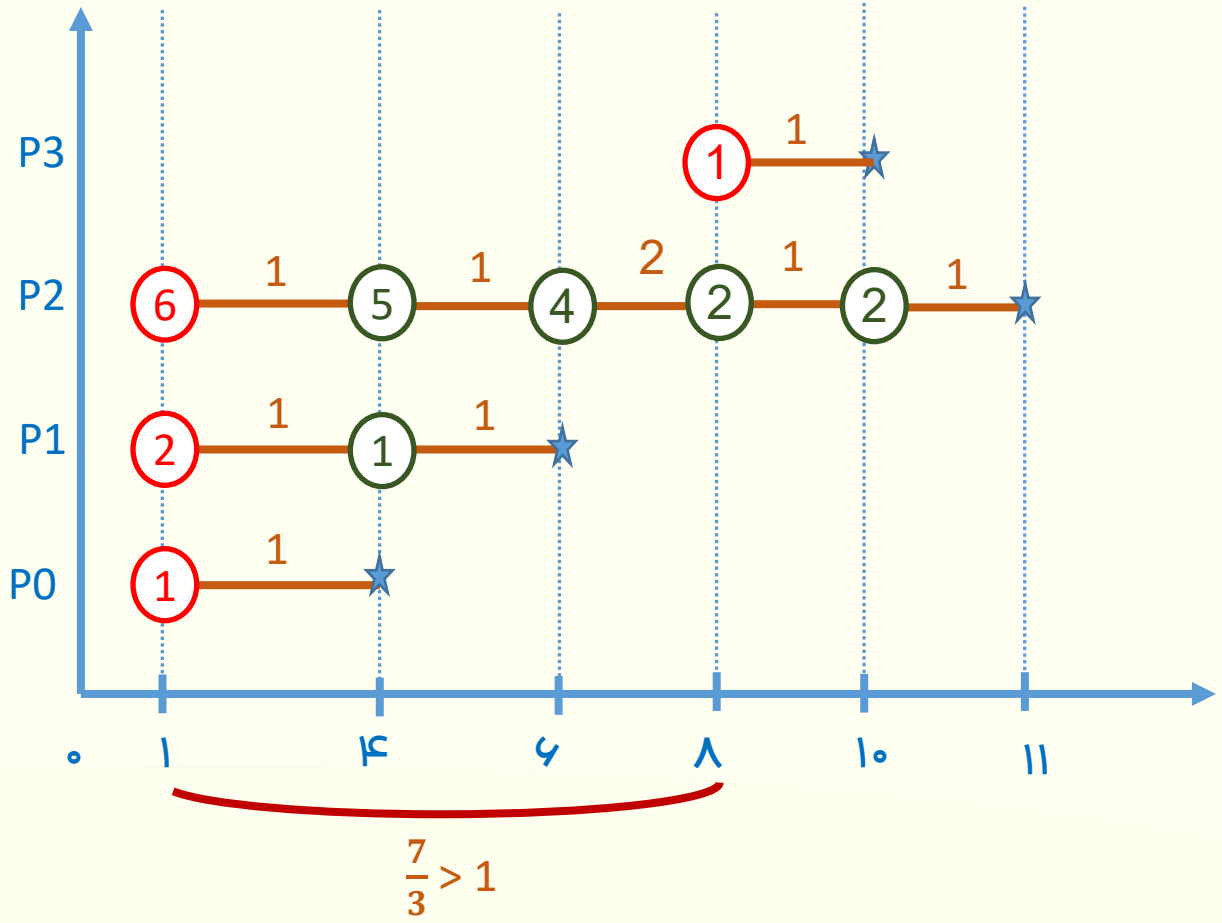
ART = (۴ + ۱۷ + ۱۳ + ۱۴ + ۸) / ۵ = ۵۶ / ۵ = ۱۱.۲

AWT = (۱ + ۱۱ + ۹ + ۹ + ۶) / ۵ = ۳۶ / ۵ = ۷.۲



مثال دوم برای حالت دوم روش RR

Process	AT	CBT	FT	RT	WT
P0	۱	۱	۴	۳	۲
P1	۱	۲	۶	۵	۳
P2	۱	۶	۱۱	۱۰	۴
P3	۸	۱	۱۰	۲	۱



نمونه سوال

نمونه سوال برای حالت دوم روش RR

فرض کنید پنج پردازه مطابق جدول زیر وارد سیستم می‌شوند. با فرض برش زمانی برابر یک میلی ثانیه مطلوب است :

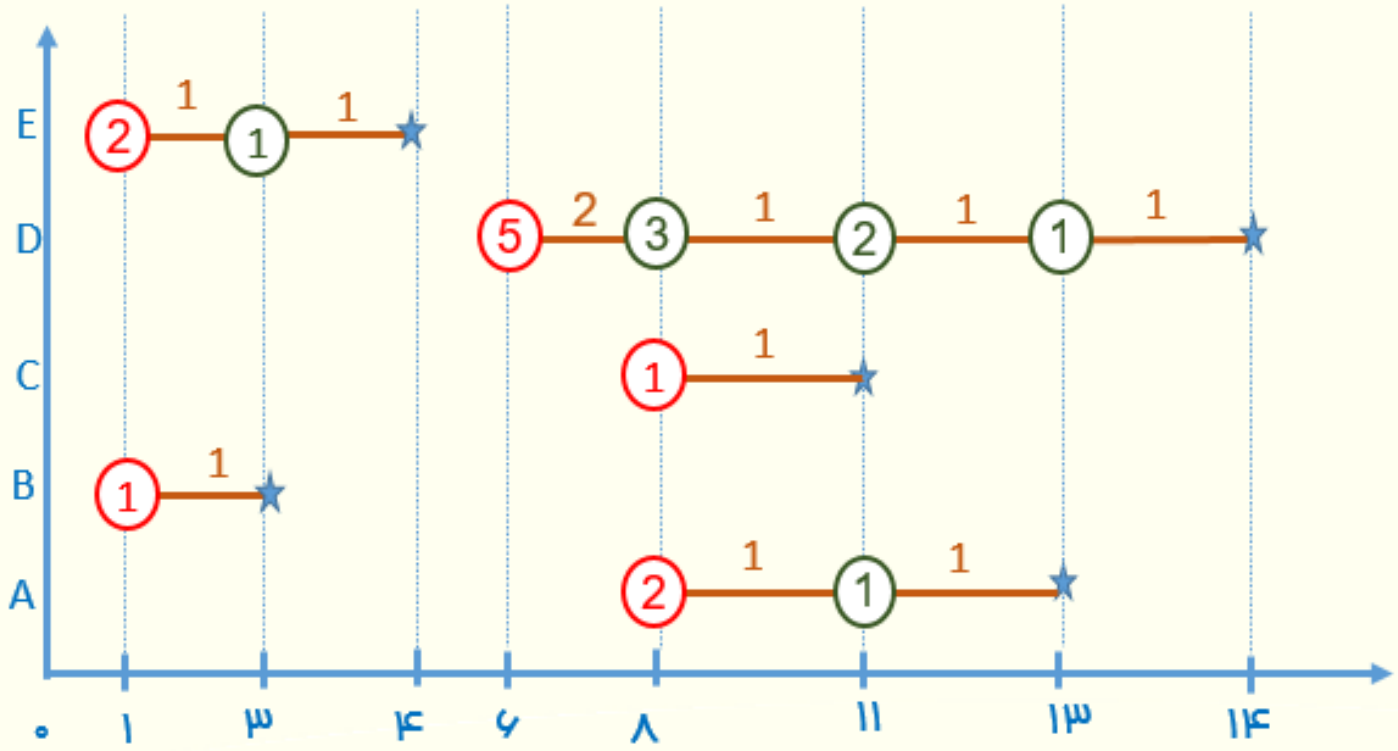
- الف) متوسط زمان پاسخدهی
- ب) متوسط زمان انتظار
- ج) مدت زمان بیکاری پردازنده
- د) درصد کارایی پردازنده

Process	AT	CBT
A	۸	۲
B	۱	۱
C	۸	۱
D	۶	۵
E	۱	۲

جواب سوال

جواب سوال مربوط به حالت دوم

Process	AT	CBT	FT	RT	WT
A	۸	۲	۱۳	۵	۳
B	۱	۱	۳	۲	۱
C	۸	۱	۱۱	۳	۲
D	۶	۵	۱۴	۸	۳
E	۱	۲	۴	۳	۱



ART = (۵ + ۲ + ۳ + ۸ +۳) / ۵ = ۲۱ / ۵ = ۴.۲

CPU idle Time = ۳

AWT = (۳ + ۱ + ۲ + ۳ + ۱) / ۵ = ۱۰ / ۵ = ۲

CPU performance = (۱۴ – ۳) / ۱۴ * ۱۰۰ = ۷۸.۵۷ %

الگوریتم ششم : کوتاه‌ترین زمان باقیمانده (SRTF - Shortest Remaining Time First)

- این روش نیز یک روش غیر انحصاری است. در این روش، از بین پردازه‌های موجود در صف ready ، پردازه‌ای را انتخاب می‌کنیم که باقیمانده زمان مورد نیازش از همه کمتر باشد. وقتی پردازه‌ای به حالت اجرا می‌رود، پردازنده را تا زمان ورود پردازه جدید در اختیار دارد. وقتی پردازه جدید وارد می‌شود، اگر زمان پردازش پردازه جدید از باقیمانده زمان مورد نیاز پردازه در حال اجرا کمتر باشد، باید جایگزین آن گردد. در غیر این صورت، پردازه در حال اجرا به کار خود ادامه می‌دهد و هیچ سوئیچی انجام نمی‌شود.

- نکته ۱ : این روش حالت غیر انحصاری روش SJF است.

- نکته ۲ : اگر زمان ورود همه پردازه‌ها یکسان باشد، عملکرد روش همانند روش SJF خواهد بود.

- نکته ۳ : به محض ورود پردازه، روش به صورت انحصاری و همانند SJF عمل می‌کند.

- نکته ۴ : ترتیب و نحوه قرار گرفتن پردازه در صف، همانند روش Round Robin است.

- نکته ۵ : اگر نیاز دو یا چند پردازه موجود در صف برابر باشد پردازه‌ای انتخاب می‌شود که زودتر وارد صف شده باشد.

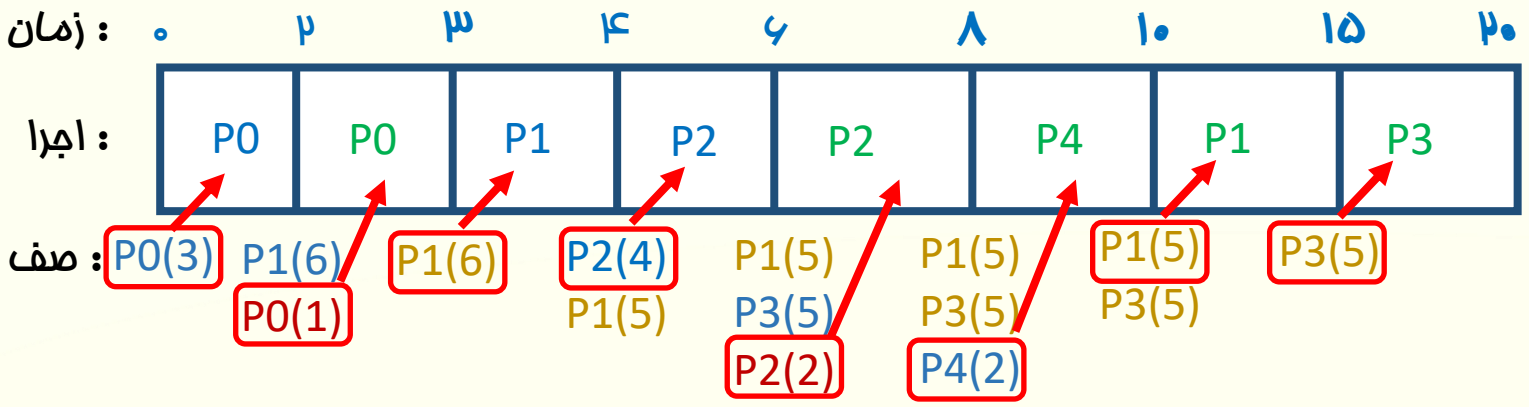
حل مثال یک

مثال حل شده برای روش SRTF

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۳	۰
P1	۲	۶		۱۳	۷
P2	۴	۴		۴	۰
P3	۶	۵		۱۴	۹
P4	۸	۲		۲	۰

ART = (۳ + ۱۳ + ۴ + ۱۴ + ۲) / ۵ = ۳۶ / ۵ = ۷.۲

AWT = (۰ + ۷ + ۰ + ۹ + ۰) / ۵ = ۱۶ / ۵ = ۳.۲



نمونه سوال

نمونه سوال برای روش SRTF

فرض کنید پنج پردازه مطابق جدول زیر وارد سیستم می‌شوند. مطلوب است :

- الف) متوسط زمان پاسخدهی
- ب) متوسط زمان انتظار

Process	AT	CBT
A	۰	۵
B	۱	۲
C	۲	۵
D	۳	۳

جواب سوال

جواب سوال مربوط به SRTF

Process	AT	CBT	FT	RT	WT
A	۰	۵	۱۰	۱۰	۵
B	۱	۲	۳	۲	۰
C	۲	۵	۱۵	۱۳	۸
D	۳	۳	۶	۳	۰

زمان : ۰ ۱ ۲ ۳ ۶ ۱۰ ۱۵

اجرا : A | B | B | D | A | C

صف : A(5) | B(2) A(4) | A(4) C(5) B(1) | A(4) C(5) D(3) | A(4) C(5) | C(5)

ART = (۱۰ + ۲ + ۱۳ + ۳) / ۴ = ۲۸ / ۴ = ۷

AWT = (۵ + ۰ + ۸ + ۰) / ۴ = ۱۳ / ۴ = ۳.۲۵

الگوریتم هفتم : اولویت (Priority)

- این روش، هم به صورت انحصاری و هم به صورت غیرانحصاری قابل پیاده سازی است. در این روش، علاوه بر زمان ورود و زمان پردازش، اولویت پردازه‌ها نیز باید مشخص باشد. این اولویت معمولاً توسط عدد برتری که به پردازه‌ها نسبت داده می‌شود، مشخص می‌گردد. عدد برتری کمتر نشان دهنده اولویت بیشتر است، مگر این‌که در صورت مسئله عکس این مطلب بیان شده باشد.

- حالت انحصاری :
  در این حالت، از بین پردازه‌های موجود در صف، پردازه‌ای انتخاب می‌شود که اولویت بیشتری داشته باشد. وقتی پردازه، پردازنده را دراختیار می‌گیرد تا زمانی‌که کارش به پایان نرسد، پردازنده را رها نمی‌کند، حتی اگر پردازه‌ای با اولویت بیشتر وارد شود.

- حالت غیرانحصاری :
  در این حالت، از بین پردازه‌های موجود در صف، پردازه‌ای انتخاب می‌شود که اولویت بیشتری داشته باشد. اگر در حین اجرای پردازه، پردازه جدیدی با اولویت بیشتر وارد شود، پردازنده از پردازه در حال اجرا گرفته شده، در اختیار پردازه جدید قرار می‌گیرد.

- **نکته ۱ :** اگر دو یا چند پردازه‌ی موجود در صف، اولویت یکسانی داشته باشند، پردازه‌ای انتخاب می‌شود که زمان پردازش کمتری داشته باشد.

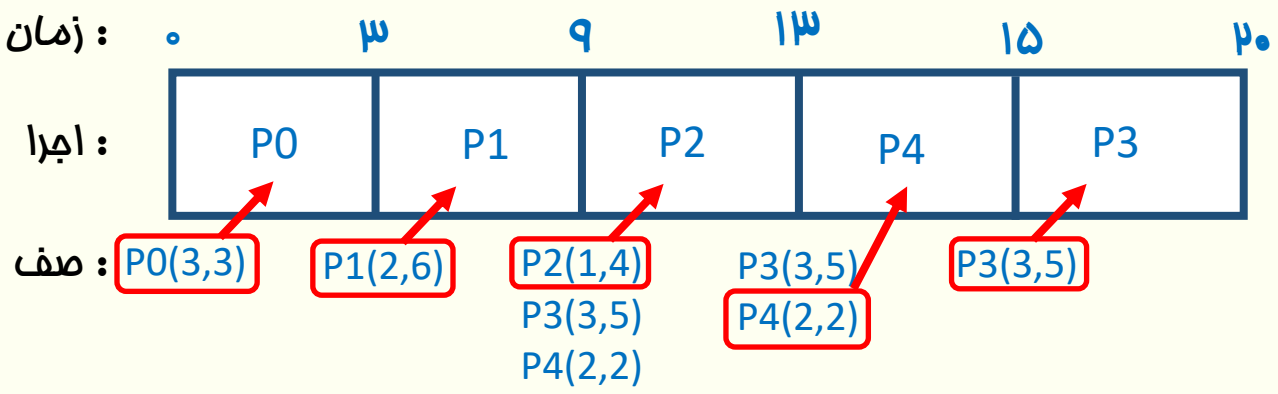
- **نکته ۲ :** اگر اولویت تمام پردازه‌ها یکسان باشد، عملکرد روش همانند روش SJF خواهد بود.

حل مثال برای حالت انحصاری

حل مثال برای حالت غیرانحصاری

مثال حل شده به روش Priority (حالت انحصاری)

Process	AT	CBT	Priority	FT	RT	WT
P0	۰	۳	۳		۳	۰
P1	۲	۶	۲		۷	۱
P2	۴	۴	۱		۹	۵
P3	۶	۵	۳		۱۴	۹
P4	۸	۲	۲		۷	۵



ART = (۳ + ۷ + ۹ + ۱۴ + ۷) / ۵ = ۴۰ / ۵ = ۸

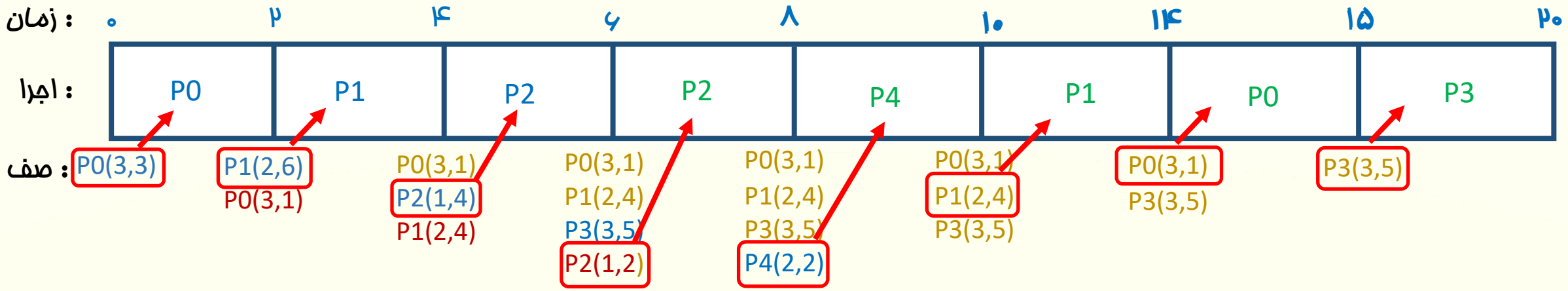
AWT = (۰ + ۱ + ۵ + ۹ + ۵) / ۵ = ۲۰ / ۵ = ۴

مثال حل شده به روش Priority (حالت غیرانحصاری)

Process	AT	CBT	Priority	FT	RT	WT
P0	۰	۳	۳		۱۵	۱۲
P1	۲	۶	۲		۱۲	۶
P2	۴	۴	۱		۴	۰
P3	۶	۵	۳		۱۴	۹
P4	۸	۲	۲		۲	۰

ART = (۱۵ + ۱۲ + ۴ + ۱۴ + ۲) / ۵ = ۴۷ / ۵ = ۹.۴

AWT = (۱۲ + ۶ + ۰ + ۹ + ۰) / ۵ = ۲۷ / ۵ = ۵.۴



الگوریتم هشتم : زمان‌بندی سه سطحی (Three Level Scheduling)

- از برخی دیدگاه‌ها، سیستم‌های دسته‌ای (Batch systems) اجازه می‌دهند زمان‌بندی در سه سطح متفاوت انجام شود :
- سطح اول : زمان‌بند پذیرش (Admission Scheduler) :

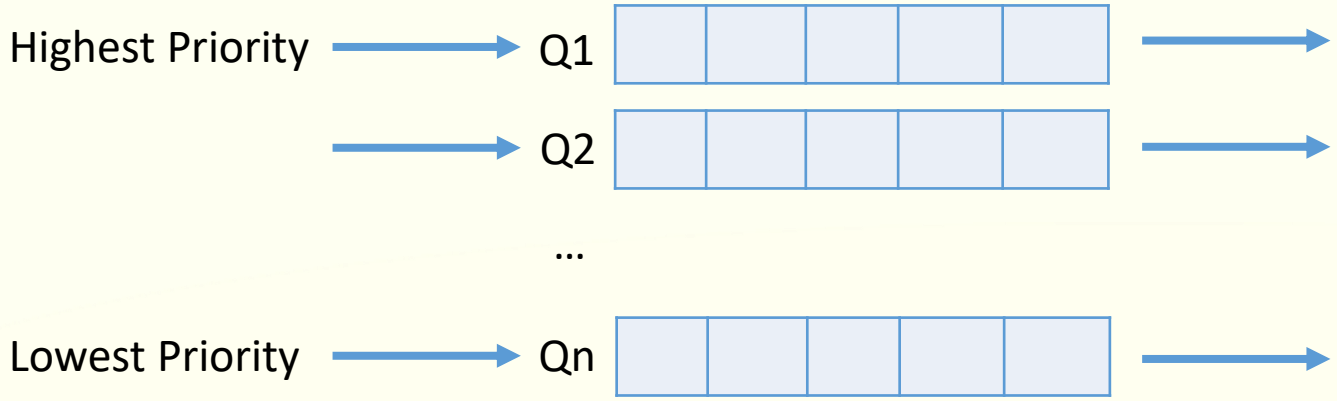
  این زمان‌بند تصمیم می‌گیرد چه کارهایی در سیستم پذیرفته شوند.
- سطح دوم : زمان‌بند حافظه (Memory Scheduler) :

  این زمان‌بند تصمیم می‌گیرد که کدام پردازه‌ها باید در حافظه اصلی بمانند و کدام پردازه‌ها به دیسک منتقل شوند.
- سطح سوم : زمان‌بند پردازنده (CPU Scheduler) :

  این زمان‌بند تصمیم می‌گیردکه کدام پردازه را از صف پردازه‌های آماده بردارد و اجرا کند.

الگوریتم نهم : صف چند سطحی (MLQ – Multi Level Queue)

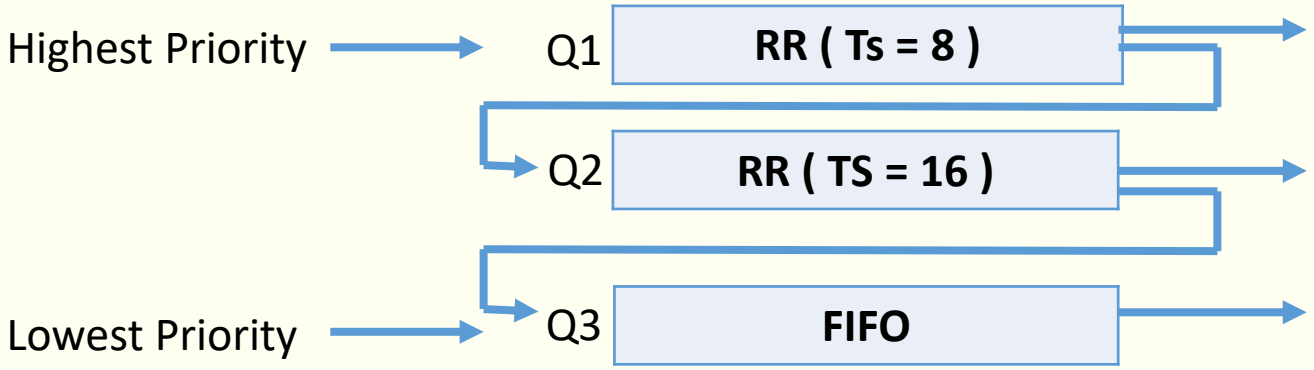
- این روش، برای مواقعی به کار می‌رود که بتوان پردازه‌ها را به راحتی در گروه‌های مختلف دسته بندی کرد.
- هر یک از گروه‌ها ممکن است زمان‌بندی متفاوتی داشته باشند. بنابراین الگوریتم‌های زمان‌بندی مختلفی نیاز خواهند داشت.
- برای هر یک از گروه‌ها صف جداگانه‌ای در نظر گرفته می‌شود.
- صف‌های بالاتر اولویت بیشتری دارند.
- هر پردازه بسته به نوع و میزان پردازش، در یکی از این صف‌ها قرار می‌گیرد.
- سیستم عامل برای انتخاب پردازه برای اجرا، ابتدا در صف با اولویت بیشتر به دنبال پردازه می‌گردد، اگر وجود داشت آن را اجرا می‌کند. در غیر این‌صورت سراغ صف با اولویت کمتر می‌رود.
- هر یک از صف‌ها می‌توانند الگوریتم زمان‌بندی خاص خود را داشته باشند.
- به هیچ عنوان پردازه ای یک صف با صف دیگر تغییر مکان نمی‌دهد.



- نکته : چون جابجایی پردازه‌ها بین صف‌ها امکان پذیر نیست، انعطاف پذیری این روش کم است.

الگوریتم دهم : صف بازخورد چند سطحی (MLFQ – Multi Level Feedback Queue)

- این روش همانند روش MLQ است با این تفاوت که پردازه‌ها ممکن است از صف با اولویت بیشتر به صف با اولویت کمتر منتقل شوند.
- در این روش تمام صف‌ها به جز پایین ترین صف، از الگوریتم RR استفاده می‌کنند. در صف آخر هر الگوریتمی قابل اعمال است.
- تفاوت بین الگوریتمهای RR موجود در صف‌های مختلف، میزان برش زمانی آن‌ها است. صف با اولویت بیشتر برش زمانی کمتری دارد و این نشان دهنده آن است که پردازههای محاوره‌ای و IO Bound در صف‌های بالاتر انجام می‌شوند و پردازه‌های CPU Bound در صف‌های پایین‌تر اجرا می‌شوند.
- مثال :



- نکته : از بالا به پایین اولویت صف‌ها کاهش می‌یابد و تا زمانی‌که پردازهای در صف بالاتر باشد، هرگز پردازه‌ای از صف پایین‌تر اجرا نخواهد شد.

حل مثال یک

مثال حل شده به روش MLFQ

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۳	۰
P1	۲	۶		۱۵	۹
P2	۴	۴		۱۴	۱۰
P3	۶	۵		۱۴	۹
P4	۸	۲		۶	۴

ART = (۳ + ۱۵ + ۱۴ + ۱۴ + ۶) / ۵ = ۵۲ / ۵ = ۱۰.۴

AWT = (۰ + ۹ + ۱۰ + ۹ + ۴) / ۵ = ۳۲ / ۵ = ۶.۴

Time	Ready Q1	Ready Q2	ReadyQ3	Run Q1	Run Q2	RunQ3	FINISH
۰	P0(3)			P0(3)			
۱		P0(2)			P0(2)		
۲	P1(6)	P0(1)			P0(1)		
۳	P1(6)			P1(6)			P0
۴	P2(4)	P1(5)		P2(4)			
۵		P2(3), P1(5)			P1(5)		
۶	P3(5)	P2(3), P1(4)			P1(4)		
۷	P3(5)	P2(3)	P1(3)	P3(5)			
۸	P2(4)	P3(4), P2(3)	P1(3)	P4(2)			
۹		P4(1), P3(4), P2(3)	P1(3)		P2(3)		
۱۱		P4(1), P3(4)	P2(1), P1(3)		P3(4)		
۱۳		P4(1)	P3(2), P2(1), P1(3)		P4(1)		
۱۴			P3(2), P2(1), P1(3)			P1(3)	P4
۱۷			P3(2), P2(1)			P2(1)	P1
۱۸			P3(2)			P3(2)	P2
۲۰							P3

الگوریتم یازدهم : تضمین (Guaranteed)

- این روش، یک روش کاملاً متفاوت برای زمان‌بندی است. در این روش، به کاربران قول‌هایی در مورد کارایی سیستم داده می‌شود.
- یکی از مواردی که ضمانت می‌شود این است که اگر n کاربر وارد سیستم شوند و با هم مشغول کار باشند، هر یک از کاربران به اندازه 1/n زمان پردازنده را در اختیار خواهد داشت.
- به‌طور مشابه، در یک سیستم تک کاربره با n پردازه در حال اجرا، تمام پردازه‌ها به یک میزان و به اندازه 1/n زمان پردازنده می‌توانند آن را در اختیار داشته باشند.
- برای این که الگوریتم به درستی پیاده سازی شود، سیستم باید مدت زمانی را که هر پردازه (از لحظه ایجاد تا کنون) از پردازنده استفاده کرده است، در جایی نگه دارد. سپس سهم هر پردازه را نیز محاسبه کند. از آنجایی که میزان استفاده واقعی هر پردازه از پردازنده مشخص است، سیستم نسبت مصرف واقعی را به سهم هر پردازه محاسبه می‌کند. در نهایت پردازه‌ای انتخاب می‌شود که کمترین نرخ مصرف را داشته باشد.
- نرخ مصرف برابراست با میزان استفاده واقعی از پردازنده تا کنون تقسیم بر سهم پردازه .

الگوریتم دوازدهم : شانسی (Lottery)

- روش ضمانت دادن به کاربران ایده خوبی است ولی پیاده سازی آن مشکل است.
- روش شانسی، با پیاده سازی خیلی راحت‌تر نسبت به روش ضمانت، نتایج قابل پیش بینی مشابهی را حاصل می‌نماید.
- ایده این روش به این صورت است که برای دستیابی به منابع مختلف نظیر وقت پردازنده، به پردازه‌ها بلیط‌های قرعه کشی (بخت آزمایی) داده می‌شود.
- هرگاه بخواهیم یک زمان‌بندی انجام دهیم به طور تصادفی یک بلیط را انتخاب می‌کنیم و منبع مورد نظر را به پردازه‌ای که بلیط را در اختیار دارد می‌دهیم.
- پردازه‌های مهم‌تر می‌توانند بلیط‌های فوق العاده دریافت کنند تا شانس برنده شدنشان افزایش یابد.
- در اینجا قانون ساده‌ای داریم : هر پردازه‌ای که کسری (f) از بلیط‌ها را در اختیار دارد، در حدود f درصد از منبع درخواستی را بدست خواهد آورد.
- **ویژگی‌های الگوریتم شانسی :**
  - اگر پردازه جدیدی وارد شود و تضمین شود که تعدادی بلیط به او داده می‌شود، به نسبت بلیط‌هایی که در اختیار دارد شانس برنده شدن خواهد داشت.
  - پردازه‌های مرتبط با هم در صورت تمایل می‌توانند بلیط‌هایشان را با هم مبادله کنند.
  - الگوریتم شانسی برای حل مسائلی به کار می‌رود که حل آن‌ها با روش‌های دیگر مشکل باشد.

- **مثال :** وقتی یک پردازه client پیامی را به یک پردازه server ارسال کرده و سپس block می‌شود، ممکن است برای افزایش شانس اجرای پردازه server ، تمام بلیط‌های خود را به پردازه server بدهد. زمانی که پردازه server به پایان می‌رسد بلیط‌های مربوط به پردازه client را به او بازمی‌گرداند

الگوریتم سیزدهم : زمان‌بندی با سهم عادلانه (FSS – Fair Share Scheduling)

- در این روش، فرض می‌کنیم هر پردازه بدون توجه به اینکه مالکش کیست، زمان‌بندی می‌شود. در نتیجه، اگر کاربر یک تعداد ۹ پردازه و کاربر دو فقط ۱ پردازه را شروع کنند، با استفاده از روش نوبت چرخشی (RR) و یا اولویت‌های یکسان، کاربر یک به میزان ۹۰ درصد و کاربر دو فقط به میزان ۱۰ درصد زمان پردازنده را در اختیار دارند.

- برای اجتناب از چنین وضعیتی به هر کاربر کسری از پردازنده اختصاص داده می‌شود و زمان‌بند به طریقی پردازه‌ها را برای اجرا انتخاب می‌کند. بنابراین اگر دو کاربر داشته باشیم که سهم هر یک ۵۰ درصد باشد، هر یک به همین میزان از پردازنده استفاده خواهند کرد و اهمیتی ندارد که هر کاربر چند پردازه دارد.

- **مثال :** فرض کنید سیستمی با دو کاربر داریم و هر یک از کاربران ۵۰ درصد سهم دارند. کاربر یک چهار پردازه A ، B ، C و D و کاربر دو فقط پردازه E را دارد. اگر از زمان‌بندی نوبت چرخشی استفاده کنیم، یکی از ترتیب‌های ممکن زمان‌بندی به صورت زیر می‌باشد:

A, E, B, E, C, E, D, E, A, E, B, E, C, E, D, E, ...

حال اگر کاربر یک دو برابر کاربر دو سهم داشته باشد داریم:

A, B, E, C, D, E, A, B, E, C, D, E, ...

اگر کاربر یک چهار برابر کاربر دو سهم داشته باشد داریم :

A, B, C, D, E, A, B, C, D, E, ...

- این روش زمانی مطرح می‌شود که در یک سیستم چندکاربره، کارهای یک کاربر بتواند به صورت چندپردازه‌ای (یا چندنخی) سازماندهی شود.

الگوریتم چهاردهم : زمان‌بندی بلادرنگ (Real Time Scheduling)

- محاسبات بلادرنگ را می‌توان به عنوان نوعی محاسبه تعریف کرد که در آن صحت و درستی سیستم علاوه بر اینکه به نتایج منطقی محاسبات بستگی دارد، به مدت زمان بدست آوردن نتایج نیز وابسته است.
- در سیستم‌های بلادرنگ، برخی پردازه‌ها یا وظیفه‌ها بلادرنگ بوده و دارای درجه‌ای از فوریت هستند. هنگام مرتبط کردن یک مهلت زمانی (Deadline) به یک وظیفه خاص ممکن است، ممکن است چنین وظیفه‌ای به دو دسته سخت و نرم تقسیم شود :
  - **وظیفه بلادرنگ سخت :**

    این وظیفه باید مهلت زمانی تعیین شده را رعایت کند. در غیر این صورت، باعث خسارت و ضرر ناخواسته و یا یک خطای مهلک خواهد بود.
  - **وظیفه بلادرنگ نرم :**

    این وظیفه نیز دارای مهلت زمانی است و رعایت آن خوب است ولی الزامی نیست. به طوری که پس از پایان مهلت زمانی نیز می‌تواند کار خود را ادامه دهد.
- رویدادهایی که سیستم‌های بلادرنگ به آنها پاسخ می‌دهد به دو دسته تقسیم می‌شوند:
  - دوره‌ای (متناوب) : این رویدادها در فواصل زمانی معین تکرار می‌شوند.
  - غیردوره‌ای (نامتناوب) : این رویدادها به صورت دوره‌ای تکرار نمی‌شوند.
- نکته : اگر m رویداد متناوب وجود داشته باشد، به‌طوری که رویداد i ام در فواصل زمانی Pi رخ دهد و به اندازه Ci به پردازنده نیاز داشته باشد، می‌گوییم رویدادها بلادرنگ هستند اگر :
- $\sum_{i=1}^{m} \frac{Ci}{Pi} \leq 1$
- در این صورت می‌گوییم سیستم قابل زمان‌بندی است.

مثال

مثال حل شده به روش Real Time

- **مثال :** یک سیستم بلادرنگ نرم را با سه رویداد متناوب با دوره‌های ۱۰۰ ، ۲۰۰ و ۵۰۰ میلی ثانیه در نظر بگیرید. اگر رویدادها به ترتیب ۵۰ ، ۳۰ و ۱۰۰ میلی ثانیه به پردازنده نیاز داشته باشند، آیا سیستم قابل زمان‌بندی است؟

- **حل :**

$$\sum_{i=1}^{m} \frac{Ci}{Pi} \leq 1$$

$$\sum_{i=1}^{3} \frac{Ci}{Pi} = \frac{50}{100} + \frac{30}{200} + \frac{100}{500} = 0.5 + 0.15 + 0.2 = 0.85 < 1$$

بنابراین سیستم قابل زمان‌بندی است.

الگوریتم پانزدهم : زمان‌بندی با مهلت زمانی (Deadline Scheduling)

- در این روش، علاوه بر زمان ورود و زمان پردازش پردازه‌ها، باید مهلت زمانی هر یک را نیز داشته باشیم.
- این روش به دو حالت قابل پیاده سازی است :

زمان‌بندی با اولویت ثابت

زمان‌بندی زودترین مهلت

- قبل از بررسی هر یک از حالت‌های فوق ابتدا مثالی مطرح کرده، سپس هر حالت را با مثال مربوطه مطرح می‌کنیم.
- مثال : فرض کنید دو پردازه A و B را داریم، به طوری که پردازه A از پنج وظیفه A1 ، A2 ، A3 ، A4 و A5 و پردازه B از دو وظیفه B1 و B2 تشکیل شده است. جدول زیر اطلاعات این وظیفه‌ها را نشان می‌دهد. ترتیب اجرا را در هر یک از حالت‌های فوق مشخص کنید.

پردازه	زمان ورود	زمان اجرا	مهلت زمانی پایان
A1	۰	۱۰	۲۰
A2	۲۰	۱۰	۴۰
A3	۴۰	۱۰	۶۰
A4	۶۰	۱۰	۸۰
A5	۸۰	۱۰	۱۰۰
B1	۰	۲۵	۵۰
B2	۵۰	۲۵	۱۰۰

حالت اول : زمان‌بندی با اولویت ثابت (Fixed Priority Scheduling)

- در این حالت، باید به اولویت پردازه‌ها توجه داشت و پردازه‌ای را انتخاب و اجرا کرد که اولویت بیشتری دارد. البته ممکن است این امر باعث شود که پردازه با اولویت کمتر در مهلت زمانی معین به پایان نرسد.

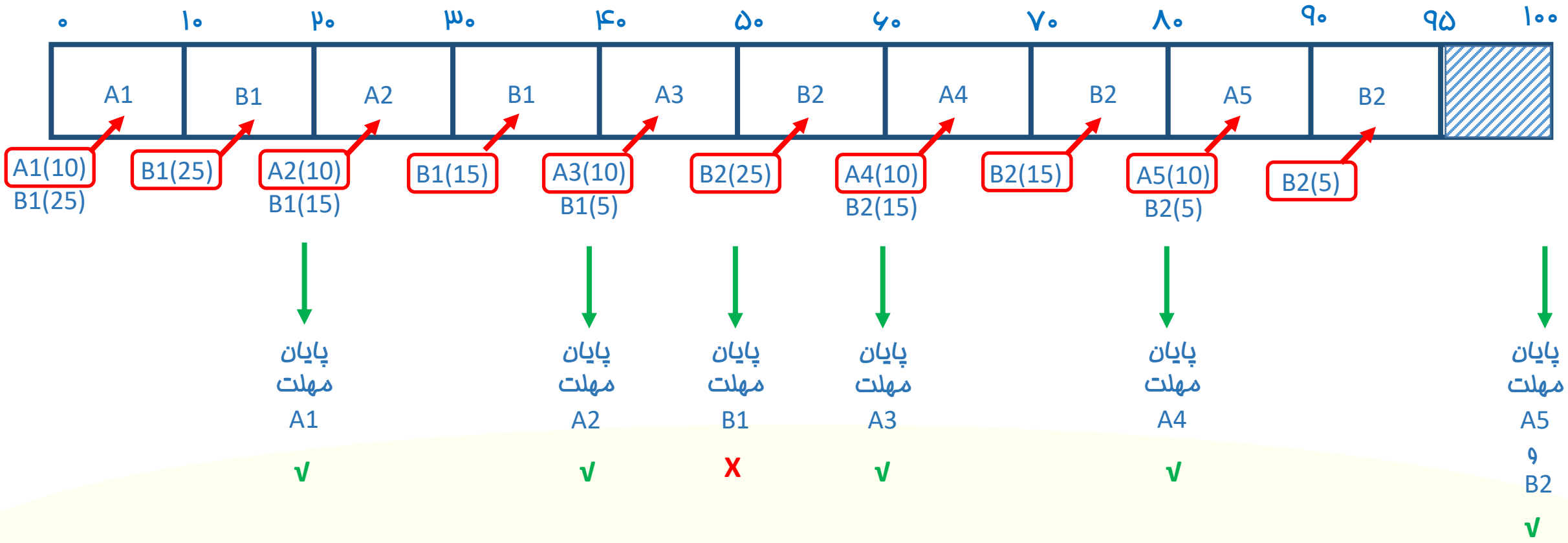
- برای حل مثال دو حالت را در نظر می‌گیریم :

  - ❖ حالت الف : اولویت A بیشتر است
  - ❖ حالت ب : اولویت B بیشتر است.

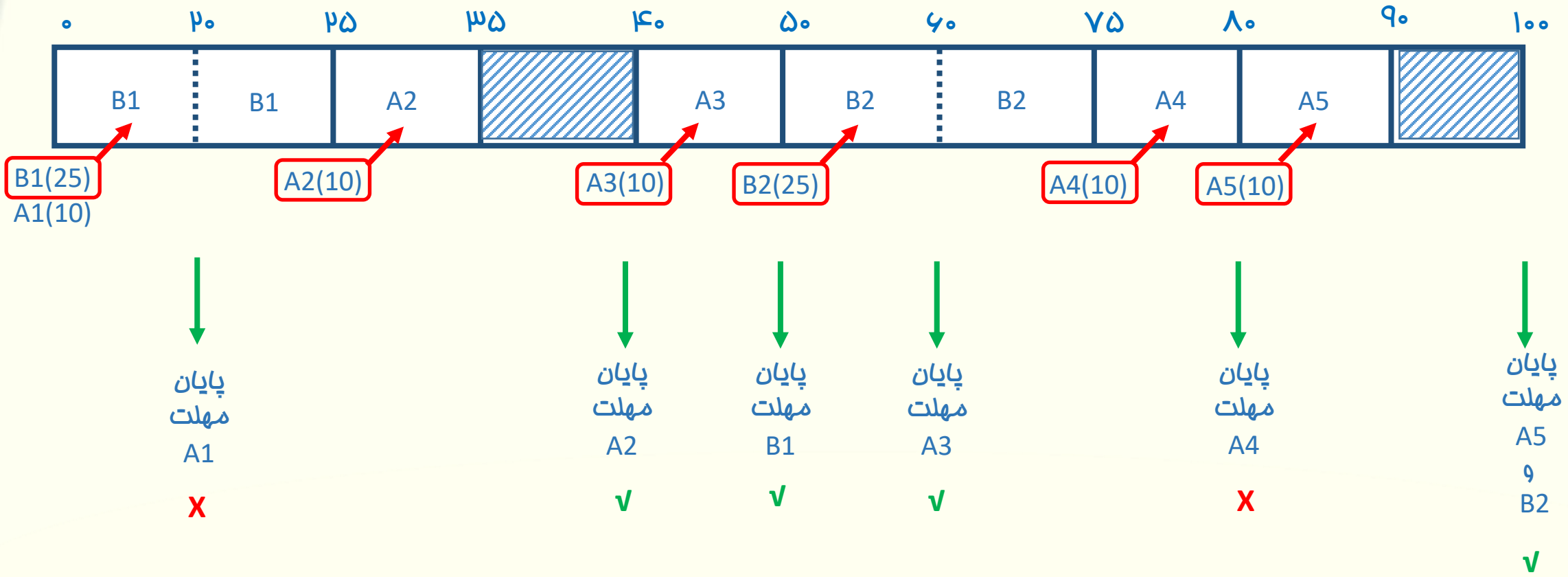
حالت الف : اولویت A بیشتر است

حالت ب : اولویت B بیشتر است

مثال برای حالت الف زمان‌بندی با اولویت ثابت

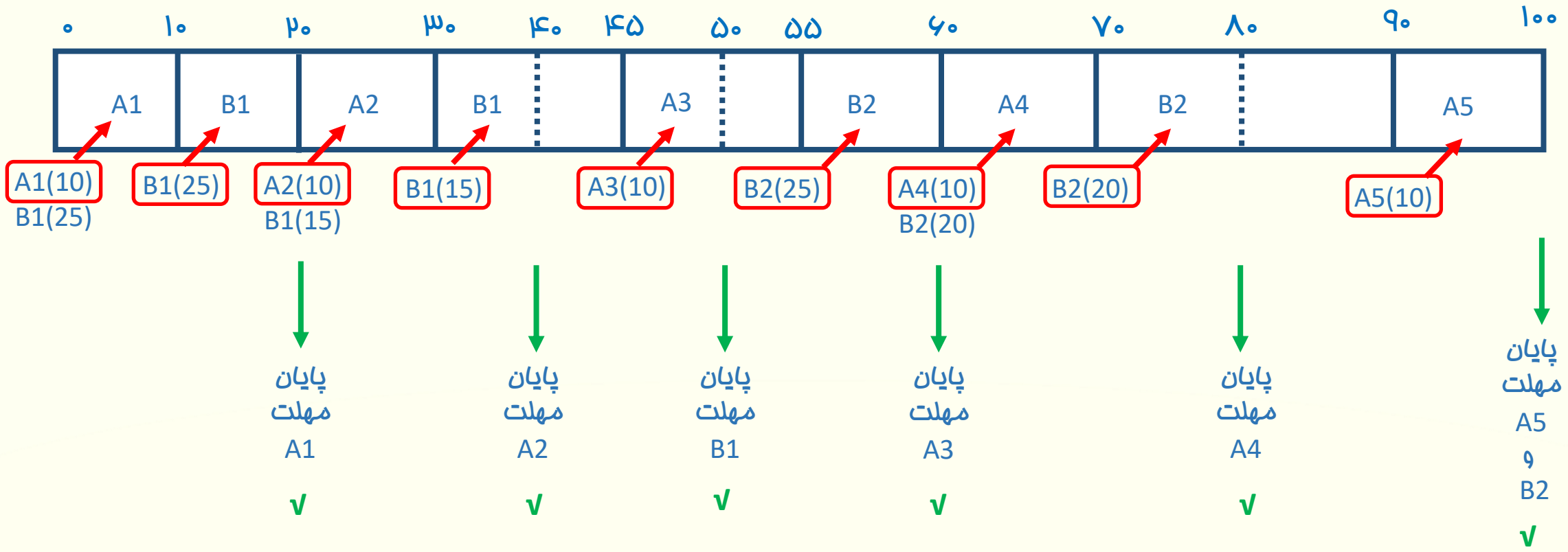


مثال برای حالت ب زمان‌بندی با اولویت ثابت



حالت دوم : زمان‌بندی زودترین مهلت (Earliest Deadline Scheduling)

- در این روش، اولویت اهمیتی ندارد بلکه برای انتخاب از بین دو یا چند پردازه (وظیفه) ، به مهلت زمانی آنها توجه می‌کنیم و وظیفه‌ای را اجرا می‌کنیم که مهلت زمانی آن نزدیک‌تر باشد.



مقایسه برخی از الگوریتمها

	FIFO	RR	SJF	SRTF	HRRN	MLFQ
حالت تصمیم گیری	انحصاری	غیرانحصاری	انحصاری	غیرانحصاری	انحصاری	غیرانحصاری
توان عملیاتی	-----	اگر برش زمانی خیلی کوچک باشد کم است	زیاد	زیاد	زیاد	-----
زمان پاسخدهی	می‌تواند زیاد باشد	برای پردازه‌های کوتاه خوب است	برای پردازه‌های کوتاه خوب است	خوب	خوب	-----
سربار	حداقل	کم	می‌تواند زیاد باشد	می‌تواند زیاد باشد	می‌تواند زیاد باشد	می‌تواند زیاد باشد
تاثیر روی پردازه‌ها	به پردازه‌های کوتاه صدمه می‌زند	عملکرد عادلانه	به پردازه‌های طولانی صدمه می‌زند	به پردازه‌های طولانی صدمه می‌زند	توازن مناسب	می‌تواند به سود پردازه‌های کوتاه باشد
گرسنگی	ندارد	ندارد	امکان دارد	امکان دارد	ندارد	امکان دارد

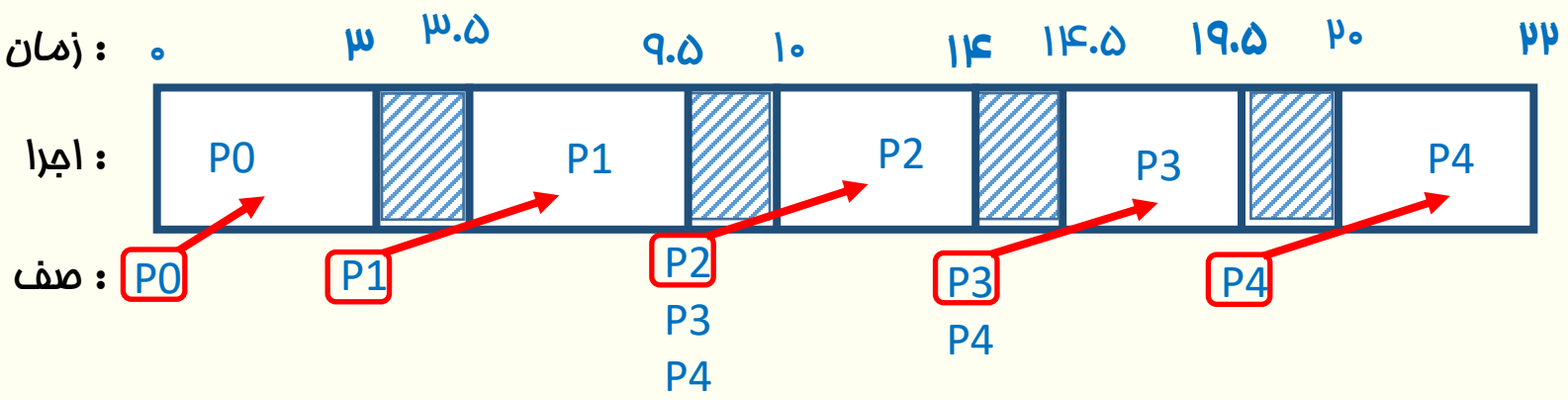
زمان تعویض متن غیر صفر

- در تمام روش‌ها و الگوریتم‌های فوق، فرض بر این بوده است که زمان تعویض متن (Context Switch) صفر در نظر گرفته شود.
- حال ممکن است مسائلی مطرح شود که در آنها این زمان صفر نباشد.
- در این صورت، زمانی که می‌خواهیم پردازه‌ای را از حالت اجرا خارج کرده و پردازه دیگری را جایگزین آن کنیم، تاخیر cs اعمال می‌شود.
- برای درک بهتر این نکته، مثال اصلی را با استفاده از روش FIFO و با فرض cs = 0.5 حل کرده و متوسط زمان پاسخدهی و متوسط زمان انتظار را بدست می‌آوریم.

حل مثال یک

مثال حل شده به روش FIFO

Process	AT	CBT	FT	RT	WT
P0	۰	۳		۳	۰
P1	۲	۶		۷.۵	۱.۵
P2	۴	۴		۱۰	۶
P3	۶	۵		۱۳.۵	۸.۵
P4	۸	۲		۱۴	۱۲



ART = (۳ + ۷.۵ + ۱۰ + ۱۳.۵ + ۱۴) / ۵ = ۴۸ / ۵ = ۹.۶

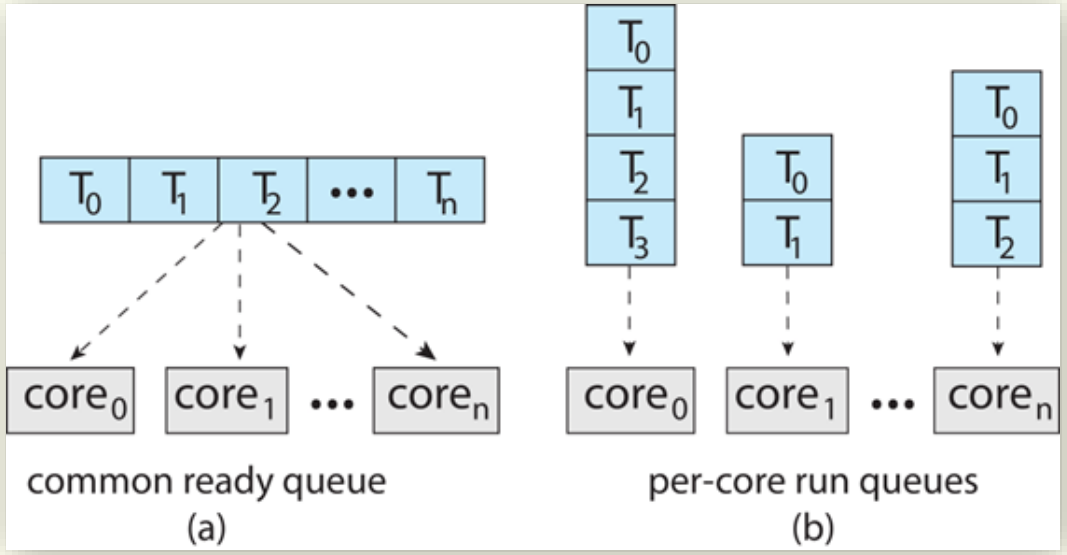
AWT = (۰ + ۱.۵ + ۶ + ۸.۵ + ۱۲) / ۵ = ۲۸ / ۵ = ۵.۶

زمان‌بندی نخ (Thread Scheduling)

- زمانی که از نخ‌ها پشتیبانی شود، زمان‌بندی روی نخ‌ها انجام می‌شود نه پردازه‌ها.
- مدل‌های چند به یک و چند به چند، کتابخانه نخ، نخ‌های سطح کاربر را برای اجرا زمان‌بندی می‌کند.
  - ❖ از آنجایی که رقابت زمان‌بندی در داخل پردازه است، این پردازه به عنوان ناحیه رقابت پردازه (PCS – Process Contention Scope) شناخته می‌شود.
  - ❖ این کار معمولا از طریق اولویت تعیین شده توسط برنامه نویس انجام می‌شود.
- نخ هسته زمان‌بندی شده روی cpu موجود که ناحیه رقابت سیستم (SCS – System Contention Scope) است، با تمام نخ های موجود در سیستم رقابت دارد.

زمان‌بندی چند پردازنده‌ای (Multiple Processor Scheduling)

- زمانی که چندین cpu در دسترس باشد، زمان‌بنید cpu پیچیده‌تر می‌شود.
- چند پردازنده ممکن است یکی از معماری‌های زیر باسد:
  - پردازنده‌های چند هسته‌ای (Multicore CPUs)
  - هسته‌های چند نخی (Multithreaded cores)
  - سیستم‌های نوما (NUMA Systems)
  - چندپردازشی ناهمگن (Heterogeneous multiprocessing)
- چندپردازشی متقارن (SMP – Symmetric multiprocessing) جایی است که در آن هر پردازنده زمان‌بندی خود را انجام می‌دهد.
  - حالت a) همه نخ‌ها ممکن است در یک صف آماده مشترک باشند.
  - حالت b) هر پردازنده ممکن است صف مخصوص نخ‌های خود را داشته باشد.

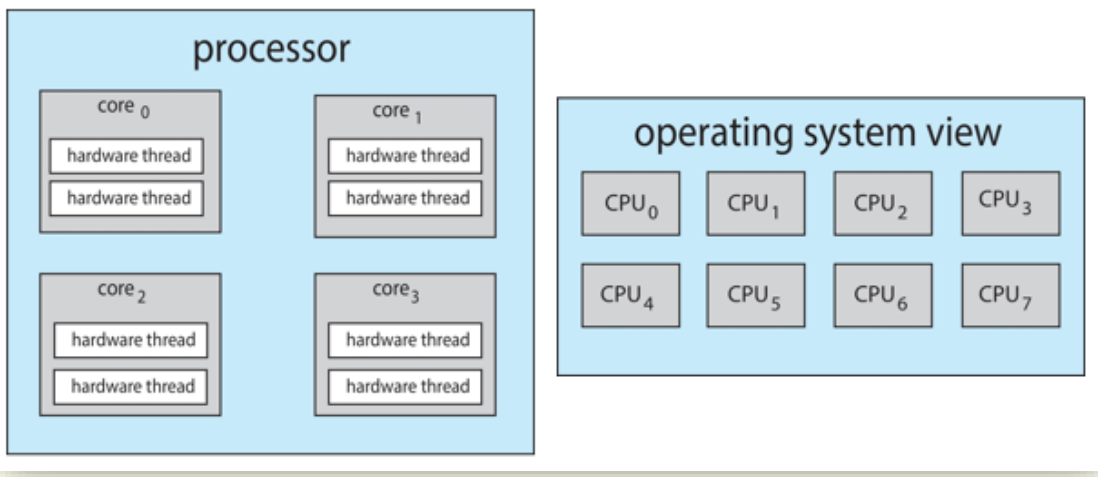


پردازنده‌های چند هسته‌ای (Multicore Processors)

- اخیراً تمایل در این است که چندین هسته پردازنده در یک تراشه فیزیکی قرار داده شوند.
- این حالت سریع‌تر است و مصرف برق کمتری دارد.
- وجود چندین نخ در هر هسته هم در حال رشد است.

سیستم چند هسته‌های چند نخی (Multithreaded Multicore System) :

- هر هسته بیش از یک نخ سخت افزاری دارد.
- اگر یک نخ مشکل memory stall داشته باشد، به نخ دیگر سوئیچ می‌کند.
  - ❖ Memory stall یعنی پردازنده برای دسترسی باید مدت زمان قابل توجهی منتظر بماند تا داده‌ها در دسترس قرار گیرند.
- تراشه چند نخی به هر هسته چندین نخ سخت افزاری اختصاص می‌دهد. در یک سیستم چهارهسته‌ای با دو نخ سخت افزاری در هر هسته، سیستم عامل ۸ پردازنده منطقی را مشاهده می‌کند.



تحقیق ۶ : زمان‌بندی در سیستم‌های عامل Linux ، Windows و Solaris چگونه است ؟ (۳ نفر)

پایان فصل سوم

مهدی دادبخش

mahdi.dadbakhsh@sharif.edu

۱۴۰۱ – ۱۴۰۲