

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

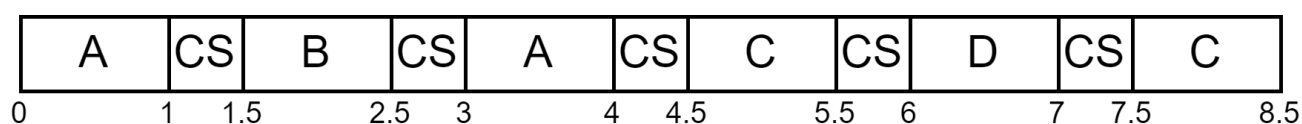
سیستم‌های عامل (بهار ۱۴۰۱)

پاسخ نامه تمرین سوم

1) با فرض استفاده از الگوریتم زمان‌بندی نوبت چرخشی (Round Robin) و برش زمانی برابر با یک و زمان تعویض فرآیند برابر با ۰/۵، ابتدا نمودار گانت را برای فرآیندهای زیر رسم کنید. سپس میانگین زمان پاسخ را محاسبه کنید.

فرآیند	لحظه‌ی ورود	زمان پردازش
A	0	2
B	0	1
C	2	2
D	4	1

زمان‌بندی با الگوریتم Round Robin به صورت زیر خواهد بود.



میانگین زمان پاسخ:

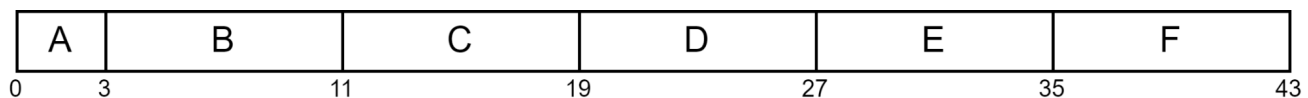
$$average\ response\ time = \frac{(0-0)+(1.5-0)+(4.5-2)+(6-4)}{4} = 1.5$$

برای مطالعه‌ی بیشتر در مورد نحوه‌ی محاسبه‌ی زمان پاسخ می‌توانید به این [لینک](#) مراجعه کنید.

2) یک سیستم تک پردازنده‌ای با صف بازخورد چند سطحی (Multi-level Feedback Queue) را در نظر بگیرید. به صف اول تکیه زمانی معادل ۸ میکروثانیه داده شده است. به سطح دوم تکیه زمانی معادل ۱۶ میکروثانیه و سطح سوم به ترتیب ورود (FCFS) زمانبندی شده است. فرض کنید ۶ کار همگی در زمان صفر به سیستم وارد می‌شوند و زمان اجرای آن‌ها به ترتیب ۳، ۸، ۱۲، ۲۰، ۲۵ و ۳۵ میکروثانیه می‌باشد. متوسط زمان برگشت (Average Turnaround Time) کارهای فوق در این سیستم چقدر خواهد بود؟ بطور کامل توضیح دهید.

در یک سیستم با صف بازخورد چند سطحی، صف‌های ابتدایی اولیوت بالاتری نسبت به صف‌های پایین‌تر دارند. وقتی یک پردازنده برای زمان‌بندی وارد سیستم می‌شود ابتدا وارد صف با بیشترین اولیت شده و اگر کار او بعد از اختصاص یک مرحله پردازنده پایان نیابد به یک صف پایین‌تر منتقل می‌شود. این اتفاق تکرار می‌شد تا در نهایت پردازنده مورد نظر پایان یابد.

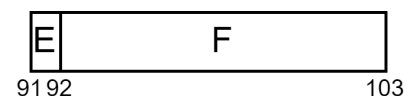
در این سیستم نیز ابتدا پردازنده‌ها وارد صف اول شده و به اندازه‌ی نهایتاً ۸ میکروثانیه پردازنده به آن‌ها تعلق می‌گیرد. پردازنده‌های A و B پایان می‌یابند اما باقی پردازنده‌ها به صف پایین‌تر منتقل می‌شوند.



در این صف نیز مشابه صف قبل به پردازنده‌ها نهایتاً ۱۶ میکروثانیه پردازنده تعلق می‌گیرد. پردازنده‌ی C و D به اتمام می‌رسند.



پردازنده‌های باقیمانده در صف نهایی به صورت FCFS به صورت زیر سرویس داده می‌شوند.



متوسط زمان برگشت:

$$\text{average turnaround time} = \frac{3+11+47+59+92+103}{6} = 52.5\mu\text{s}$$

3) پردازش‌های P1، P2 و P3 به ترتیب با زمان‌های (CPU burst) ۲ و ۴ و ۵ میلی‌ثانیه در زمان صفر وارد می‌شوند. چنانچه در آغاز اولویت آنها عکس زمان اجرایشان باشد و با گذر زمان به ازای هر ۱ میلی‌ثانیه ۰.۱ به اولویت پردازش‌های منتظر اضافه شود هر پردازش در چه زمانی به اتمام می‌رسد؟

	0					5					10	
P1	0.5	0.5										
P2	0.25	0.35	0.35	0.45	0.45	0.55	0.55	0.65	0.65			
P3	0.2	0.3	0.4	0.4	0.5	0.5	0.6	0.6	0.7	0.7	0.7	

Processes	Turn-around time	Waiting time
P1	2	0
P2	9	5
P3	11	6

4) برنامه ی زیر از Pthreads API استفاده می کند. هنگامی که زمان اجرای خط B برسد، چه تعداد نخ (thread) در سیستم به واسطه این برنامه فعال خواهد بود؟ توضیح دهید. (جواب ممکن است به صورت "a یا b" تعداد نخ باشد)

زمان اجرا شدن هر ریسمان را نمیتوانیم دقیق بیان کنیم؛ بنابراین ممکن است زمانی که دستور join اولی فراخوانی می شود در مرحله بعدی اجرای این پردازش نخ اول در نتیجه خط B اجرا شود. در این صورت دو نخ فعال داریم، یکی خود نخ اصلی که تابع main در آن فراخوانی شده و دیگری نخی که خط B را شامل می شود زیرا هنوز به پایان نرسیده است. اما اگر پس از join اول و قبل از صدا زده شدن تابع مربوط به نخ اول join دومی هم صدا زده شود و با چون نمی توانیم ترتیب اجرای نخ ها را بگوییم ممکن است نخ دومی هنوز فعال باشد. پس با احتساب نخ اصلی و نخ اولی که ایجاد کردیم سه نخ خواهیم داشت. در نتیجه دو یا سه نخ خواهیم داشت. (با تشکر از آقای بردیا اردکانیان)

```
#include <sys/types.h>
#include <sys/wait.h>
#include <pthread.h>
#include <stdio.h>
#include <unistd.h>

int value = 5;
void *runner1(void *param);
void *runner2(void *param);

int main() {
    pthread_t tid1, tid2;
    pthread_attr_t attr1, attr2;

    pthread_attr_init(&attr1);
    pthread_create(&tid1, &attr1, runner1, NULL);

    pthread_attr_init(&attr2);
    pthread_create(&tid2, &attr2, runner2, NULL);

    printf("A: value = %d\n", value);    /* LINE A */

    pthread_join(tid1, NULL);
    pthread_join(tid2, NULL);

    return 0;
}

void *runner1(void *param){
    value += 10;
    printf("B: value = %d\n", value);    /* LINE B */
    pthread_exit(0);
}

void *runner2(void *param){
    value += 10;
```

```
printf("C: value = %d\n", value);  /* LINE C */  
pthread_exit(0);  
}
```

5) متوسط زمان برگشت برای فرایندهای جدول ۱ را با هر یک از الگوریتم های زیر به دست آورید. (شامل Gantt chart) اندازه کوانتوم ۲ واحد زمانی است و فرآیند وارد شده به سیستم می تواند فرآیند موجود را قبضه کند. (preemptive)

الف) First Come First Serve (FCFS)

ب) Shortest Remaining Time First (SRTF)

پ) Shortest Job First (SJF)

ت) Round Robin (RR)

فرآیند	زمان ورود	زمان مورد نیاز (زمان پردازش)
p1	1	6
p2	4	5
p3	2	4
p4	5	11
p5	7	3

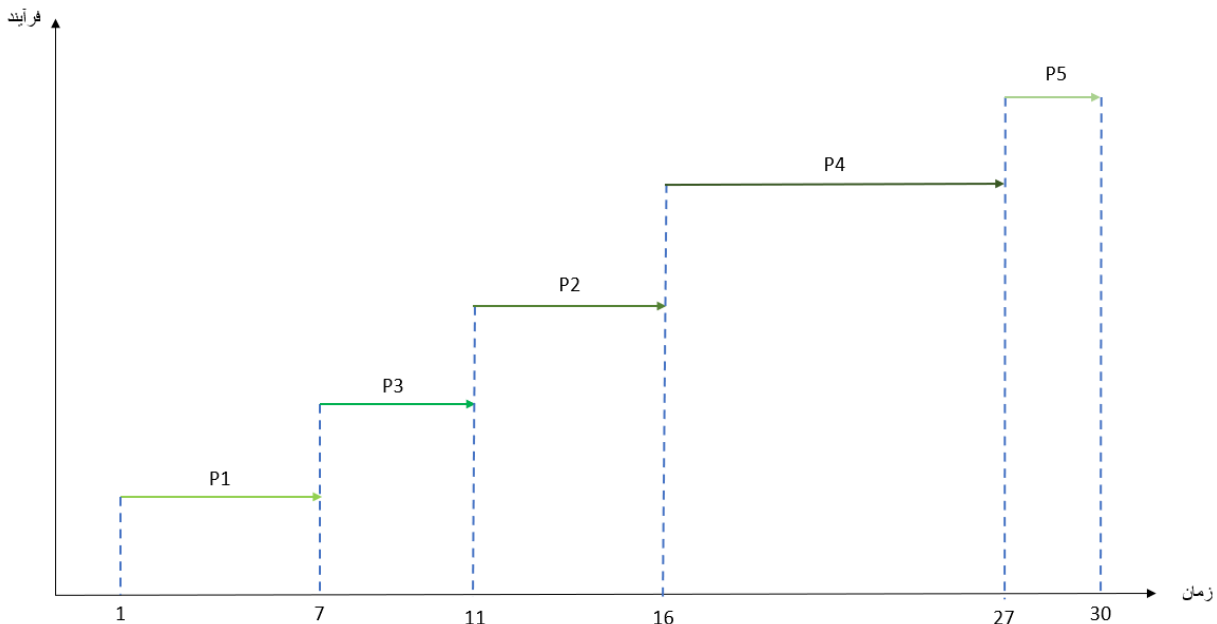
جدول ۱

الف) FCFS

الگوریتم FCFS یک الگوریتم Non Preemptive می باشد، بنابراین یک فرآیند به طور کامل اجرا می شود و سپس فرآیند بعدی وارد پردازنده می شود. شکل زیر Gantt chart زمان بندی این فرآیند ها بر اساس الگوریتم FCFS را نمایش می دهد:



همچنین نمودار زیر نحوه انتخاب فرآیند ها در این الگوریتم را نمایش می دهد:



حال با توجه به شکل بالا زمان برگشت هر یک از فرآیند ها را محاسبه می کنیم:

فرآیند	زمان ورود	زمان مورد نیاز	زمان خروج	زمان برگشت
P1	1	6	7	$7 - 1 = 6$
P2	4	5	16	$16 - 4 = 12$
P3	2	4	11	$11 - 2 = 9$
P4	5	11	27	$27 - 5 = 22$
P5	7	3	30	$30 - 7 = 23$

بنابراین میانگین زمان برگشت برای فرآیندهای عنوان شده در این الگوریتم برابر است با:

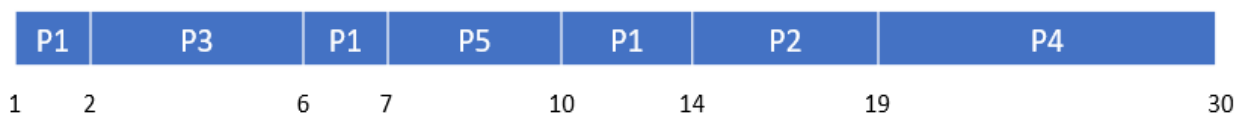
$$Average_{TurnaroundTime} = \frac{6 + 12 + 9 + 22 + 23}{5} = \frac{72}{5} = 14.4$$

$$\Rightarrow Average_{TurnaroundTime} = 14.4$$

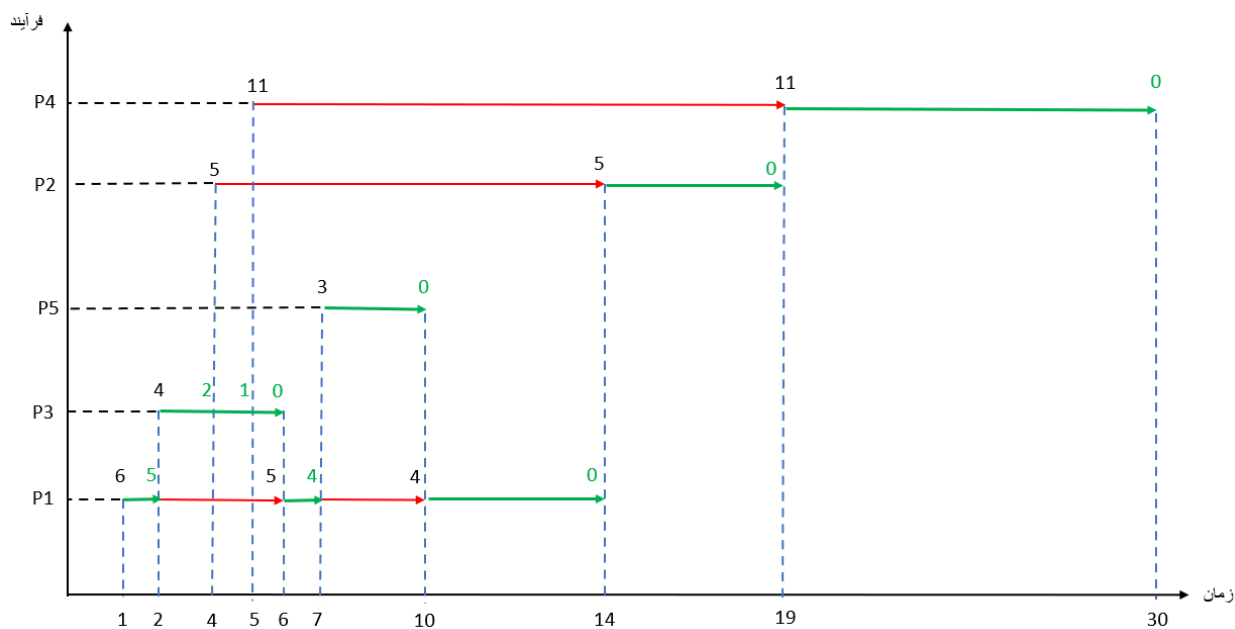
ب) SRTF

الگوریتم SRTF یک الگوریتم Preemptive می باشد، بنابراین در صورتی که فرآیندی با کار کمتر وارد شود فرآیند قبلی متوقف و فرآیند جدید اجرا می شود؛ این الگوریتم Preemptive شده الگوریتم SJF می باشد؛ شکل زیر Gantt chart زمان بندی این فرآیند ها را بر اساس الگوریتم SRTF نمایش می دهد، با توجه به اینکه در زمان 6 هر دو فرآیند P1 و P2 دارای یک میزان کار باقی مانده می باشند به دو صورت می توان Gantt chart را رسم کرد:

حالت اول:



همچنین نمودار زیر هر یک از فرآیند ها را از زمان ورود تا زمان خروج کامل نمایش می دهد که در آن خطوط سبز رنگ فرآیند های در حال اجرا و خطوط قرمز رنگ بقیه فرآیند ها را نمایش می دهد، همچنین میزان زمان مورد نیاز برای اتمام هر کار نیز در بالا هر یک از خطوط نوشته شده است و اعدادی که به رنگ سبز نوشته شده اند میزان زمان باقی مانده برای اتمام کار فرآیند در حال اجرا را نمایش می دهد:



حال با توجه به شکل بالا زمان برگشت هر یک از فرآیندها را محاسبه می کنیم:

فرآیند	زمان ورود	زمان مورد نیاز	زمان خروج	زمان برگشت
P1	1	6	14	$14 - 1 = 13$
P2	4	5	19	$19 - 4 = 15$
P3	2	4	6	$6 - 2 = 4$
P4	5	11	30	$30 - 5 = 25$
P5	7	3	10	$10 - 7 = 3$

بنابراین میانگین زمان برگشت برای فرآیندهای عنوان شده در این الگوریتم برابر است با:

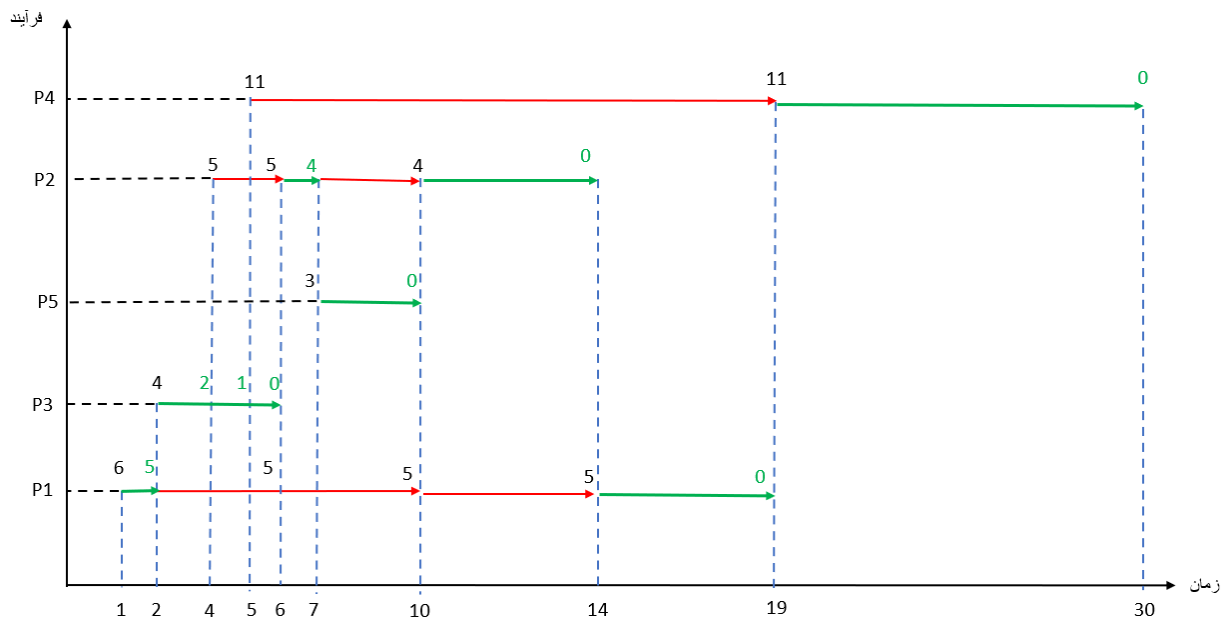
$$Average_{TurnaroundTime} = \frac{13 + 15 + 4 + 25 + 3}{5} = \frac{60}{5} = 12$$

$$\Rightarrow \text{Average}_{TurnaroundTime} = 12$$

حالت دوم:

P1	P3		P2	P5		P2	P1	P4	
1	2	6	7	10	14	19	30		

همچنین نمودار زیر نحوه انتخاب فرآیندها در این الگوریتم را نمایش می دهد:



حال با توجه به شکل بالا زمان برگشت هر یک از فرآیند ها را محاسبه می کنیم:

فرآیند	زمان ورود	زمان مورد نیاز	زمان خروج	زمان برگشت
P1	1	6	19	$19 - 1 = 18$
P2	4	5	14	$14 - 4 = 10$
P3	2	4	6	$6 - 2 = 4$
P4	5	11	30	$30 - 5 = 25$
P5	7	3	10	$10 - 7 = 3$

بنابراین میانگین زمان برگشت برای فرآیندهای عنوان شده در این الگوریتم برابر است با:

$$Average_{TurnaroundTime} = \frac{18 + 10 + 4 + 25 + 3}{5} = \frac{60}{5} = 12$$

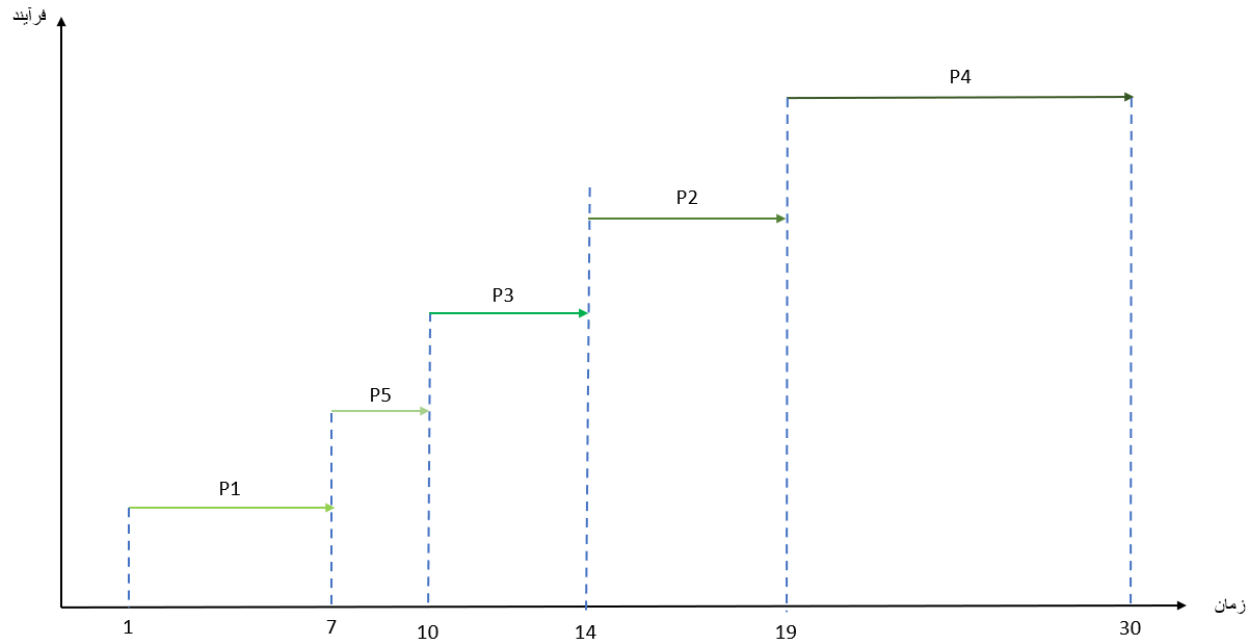
$$\Rightarrow \text{Average}_{TurnaroundTime} = 12$$

پ) SJF

الگوریتم SJF یک الگوریتم Non Preemptive می باشد، بنابراین یک فرآیند به طور کامل اجرا می شود و سپس فرآیند بعدی وارد پردازنده می شود. شکل زیر Gantt chart زمان بندی این فرآیند ها بر اساس الگوریتم SJF را نمایش می دهد:



همچنین نمودار زیر نحوه انتخاب فرآیند ها در این الگوریتم را نمایش می دهد:



حال با توجه به شکل بالا زمان برگشت هر یک از فرآیندها را محاسبه می‌کنیم:

فرآیند	زمان ورود	زمان مورد نیاز	زمان خروج	زمان برگشت
P1	1	6	7	$7 - 1 = 6$
P2	4	5	19	$19 - 4 = 15$
P3	2	4	14	$14 - 2 = 12$
P4	5	11	30	$30 - 5 = 25$
P5	7	3	10	$10 - 7 = 3$

بنابراین میانگین زمان برگشت برای فرآیندهای عنوان شده در این الگوریتم برابر است با:

$$Average_{TurnaroundTime} = \frac{6 + 15 + 12 + 25 + 3}{5} = \frac{61}{5} = 12.2$$

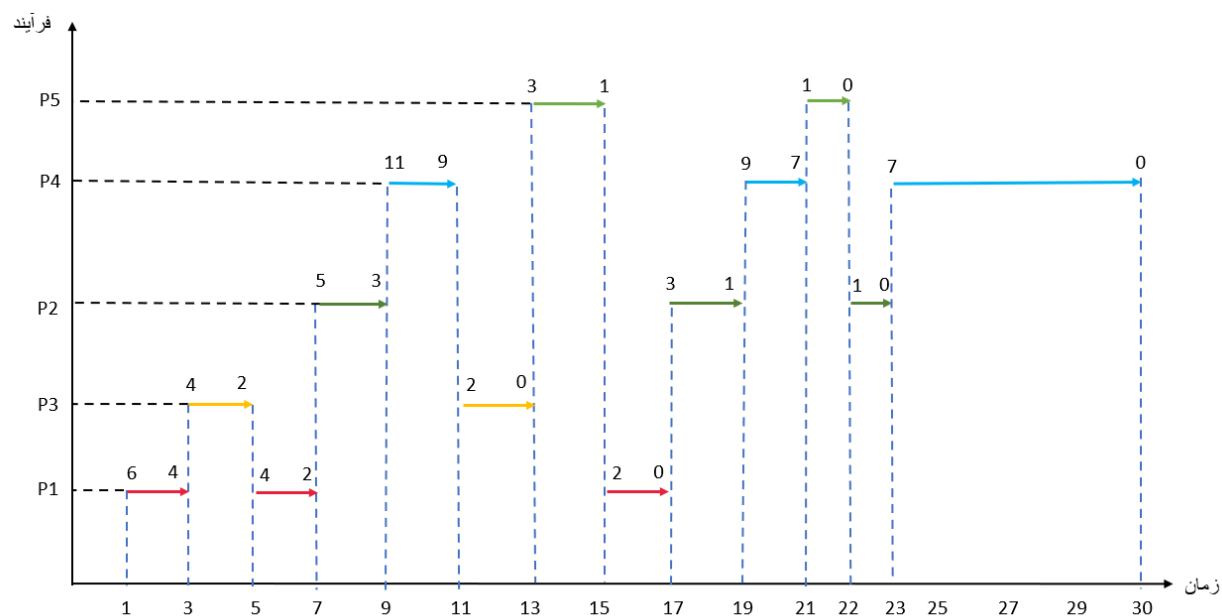
$$\Rightarrow Average_{TurnaroundTime} = 12.2$$

ت) RR

الگوریتم RR یک الگوریتم Preemptive می‌باشد، این الگوریتم در صورتی که فرآیندهای دیگری بجز فرآیند در حال اجرا در سیستم باشد در هر بازه زمانی (2 کوانتوم) فرآیند فعلی را خارج و فرآیند دیگری را به ترتیب زمان ورودش به پردازنده وارد می‌کند؛ شکل زیر Gantt chart زمان بندی این فرآیندها را بر اساس الگوریتم RR نمایش می‌دهد (با توجه به اینکه در برخی از بازه های زمانی برخی زمان ورود برخی از فرآیندها با یک دیگر برابر است این Gantt chart را به صورت های دیگری نیز رسم کرد، در اینجا فرض شده که در صورتی که زمان ورود دو فرآیند با یک دیگر یکسان باشد فرآیندی انتخاب می‌شود که قبلاً اجرا نشده است):

P1	P3	P1	P2	P4	P3	P5	P1	P2	P4	P5	P2	P4	
1	3	5	7	9	11	13	15	17	19	21	22	23	30

همچنین نمودار زیر نحوه انتخاب فرآیندها در این الگوریتم را نمایش می دهد:



حال با توجه به شکل بالا زمان برگشت هر یک از فرآیندها را محاسبه می کنیم:

فرآیند	زمان ورود	زمان مورد نیاز	زمان خروج	زمان برگشت
P1	1	6	17	$17 - 1 = 16$
P2	4	5	23	$23 - 4 = 19$
P3	2	4	13	$13 - 2 = 11$
P4	5	11	30	$30 - 5 = 25$
P5	7	3	22	$22 - 7 = 15$

بنابراین میانگین زمان برگشت برای فرآیندهای عنوان شده در این الگوریتم برابر است با:

$$Average_{TurnaroundTime} = \frac{16 + 19 + 11 + 25 + 15}{5} = \frac{86}{5} = 17.2$$

$$\Rightarrow Average_{TurnaroundTime} = 17.2$$