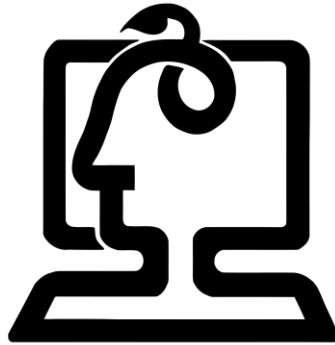


«به نام ایزد یکتا»



دانشکده مهندسی کامپیوتر

گزارش هشتم درس آزمایشگاه سیستم عامل

استاد: مهندس قاسمی

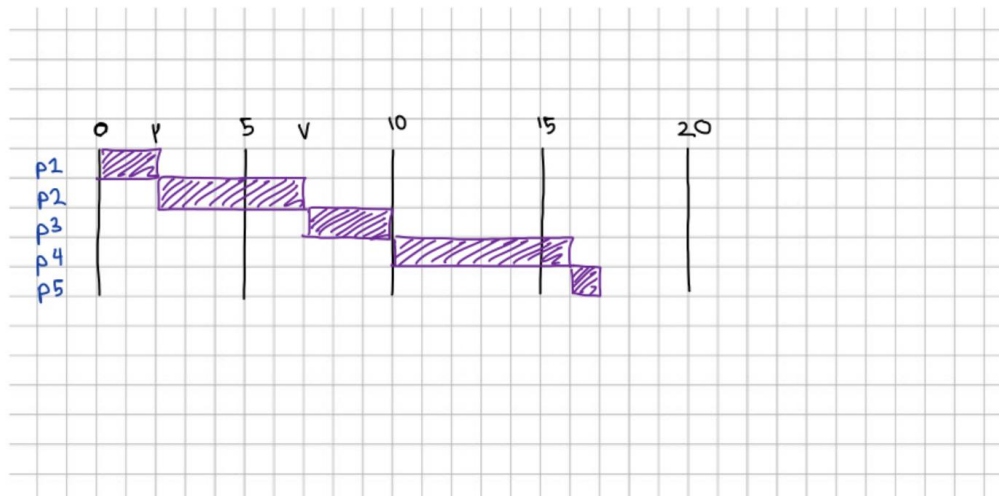
تهیه کننده: بردیا اردکانیان

۹۸۳۱۰۷۲

FCFS (بخش اول)

در این بخش با توجه به ورود پردازش‌ها، هر کدام زودتر آمده باشد (طبق ترتیب وارد کردن آن‌ها در ورودی) اجرا خواهند شد. در پیاده‌سازی، از این نکته استفاده می‌کنیم که زمان انتظار هر پردازش برابر است با زمان اجرای فرآیند قبل آن و زمان اجرای آن هم برابر است با مجموع Burst Time.

process	burst time	waiting time	turnaround time
p1	2	0	2
p2	5	2	7
p3	3	7	10
p4	6	10	16
p5	1	16	17



$$\text{avg waiting time} = \frac{0 + 2 + 7 + 10 + 16}{5} = 7$$

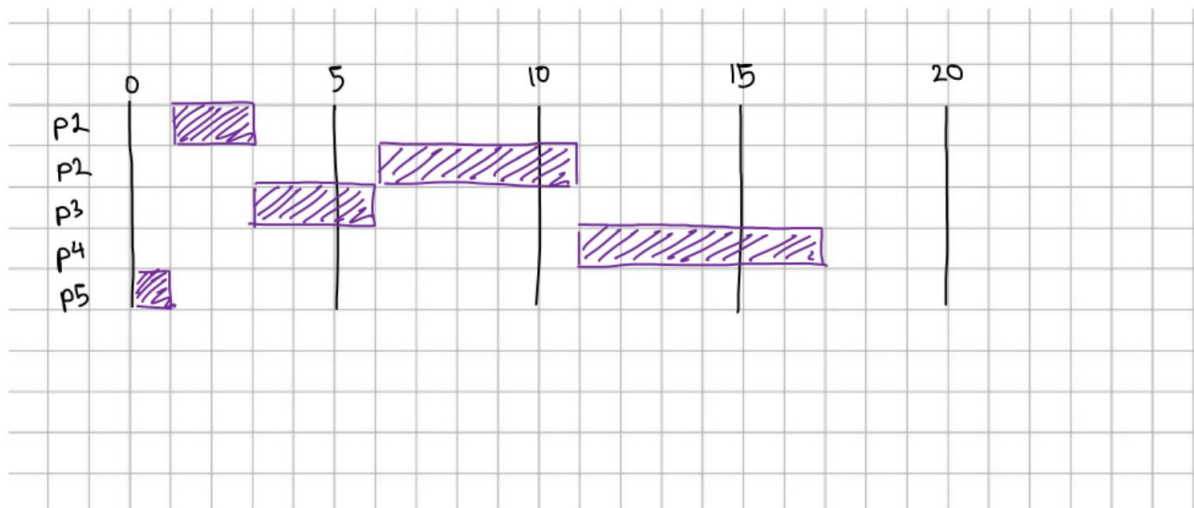
$$\text{avg turnaround time} = \frac{2 + 7 + 10 + 16 + 17}{5} = 10.4$$

شکل 1-1 (تحلیل FCFS)

بخش دوم) SJF

برای این بخش پس از گرفتن ورودی های لازم مانند بخش قبل، پردازش ها را به ترتیب صعودی **cpu burst time** سورت می کنم و سپس آن ها را اجرا می کنم. (این قسمت مانند قسمت قبلی اجرا می شود چون ترتیب پردازش ها را می دانیم).

process	burst time	waiting time	turnaround time
p1	2	1	3
p2	5	6	11
p3	3	3	6
p4	6	11	17
p5	1	0	1



$$\text{Avg turnaround time} = \frac{3 + 11 + 6 + 17 + 1}{5} = 7.4$$

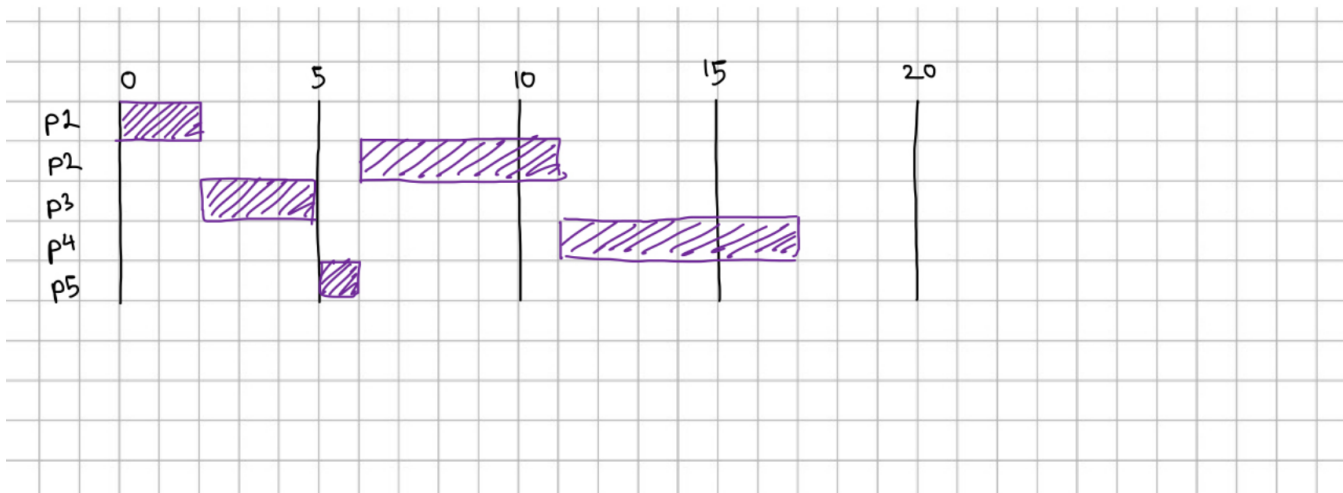
$$\text{Avg waiting time} = \frac{1 + 6 + 3 + 11 + 0}{5} = 4.2$$

شکل 2-1 (تحلیل SJF)

بخش سوم) Priority

برای این بخش پس از گرفتن ورودی های الزم مانند ۲ بخش قبل، پردازش ها را به ترتیب صعودی اولویت ها سورت می کنم زیرا هر چه عدد اولویت کمتر باشد، اولویت آن بیشتر است و سپس آن ها را اجرا می کنم. (این قسمت مانند قسمت اول اجرا می شود چون ترتیب پردازش ها را می دانیم).

process	burst time	priority	waiting time	turnaround time
p1	2	1	0	2
p2	5	5	4	11
p3	3	2	2	5
p4	6	7	11	17
p5	1	3	5	4



$$\text{Avg turnaround time} = \frac{2 + 11 + 5 + 17 + 4}{5} = 8.2$$

$$\text{Avg waiting time} = \frac{0 + 4 + 2 + 11 + 5}{5} = 4.2$$

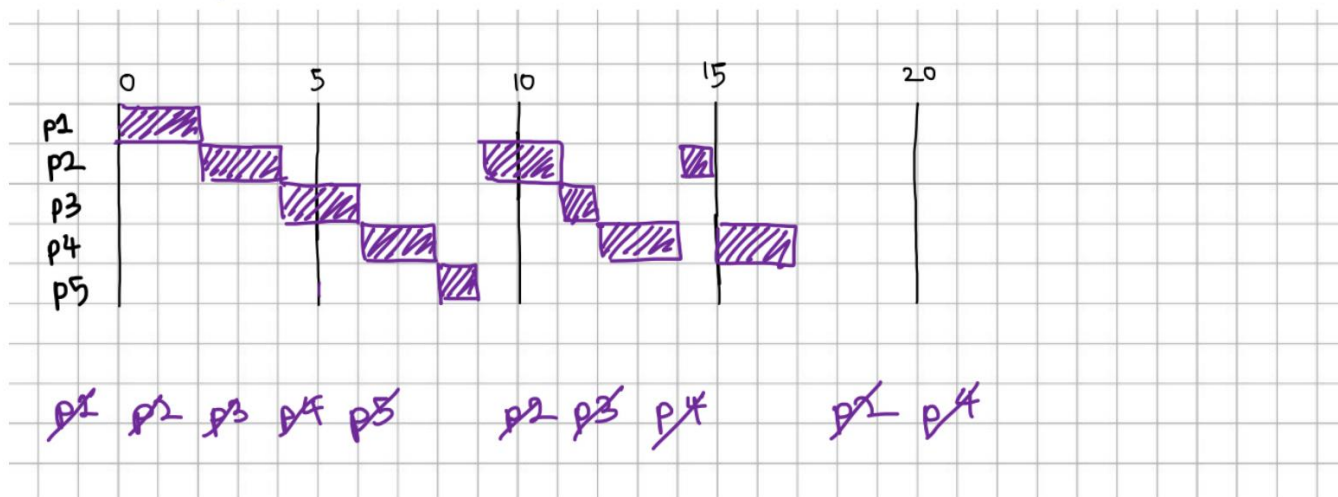
شکل 3-1 (تحلیل Priority)

بخش چهارم Round Robin

در اين بخش با گرفتن يك quantum time از پردازش مشخص مي كنيم كه هر پردازش تنها مدت زمان محدودی برای اجرا دارد و سپس نوبت پردازش بعدی می شود و این مدت زمان محدود همان quantum time است.

process	burst time	waiting time	turnaround time
p1	2	0	2
p2	8 3 1	10	15
p3	3 1	9	12
p4	8 4 2	11	17
p5	1	8	9

time quantum = 2



$$\text{Avg turnaround time} = \frac{2 + 15 + 12 + 17 + 9}{5} = 11$$

$$\text{Avg waiting time} = \frac{0 + 10 + 9 + 11 + 8}{5} = 11.4$$

شکل 4-1 (تحليل Round Robin)

بخش پنجم) مقایسه

کاربرد	الگوریتم
این روش پیاده سازی و منطق آسانی دارد ولی از طرفی برای تعداد پردازش های زیاد باعث قحطی می شود زیرا همواره پردازش های جدید را اجرا می کند و باعث می شود بعضی پردازش ها خیلی منتظر بمانند. این روش برای batch system مناسب است و برای سیستم هایی مثل صف خرید بلیط سینما کاربرد دارد.	FCFS
این روش باعث می شود throughput بالا رود زیرا پردازش های کوتاه تر را ابتدا اجرا می کند و معمولاً در سیستم تعداد پردازش های کوتاه بیشتر از طولانی است. در اینجا ولی نیز قحطی داریم و باعث می شود پردازش های طولانی هیچگاه اجرا نشوند. این روش باعث می شود Average Waiting Time کاهش یابد همانطور که در این آزمایش دیدیم که کمترین Average Waiting Time را داشت پس برای Long term scheduling مناسب است در پردازش ها با تعداد بالا. این روش برای batch system مناسب است.	SJF
این روش برای زمانی مناسب است که برای هر پردازش اولویت آن را بدانیم و در آن صورت باعث می شود پردازش ها با اهمیت بیشتر برای ما ابتدا اجرا شوند، البته در اینجا نیز احتمال قحطی وجود دارد ولی می توانیم با الگوریتم aging این موضوع را کنترل کنیم. این روش برای Interactive system مناسب است	Priority
این روش باعث می شود که turnaround ما به صورت میانگین افزایش یابد ولی به صورت کلی response time آن کمتر است. البته باید به این موضوع توجه کرد که اگر q بزرگ باشد شبیه به همان fcfs می شود و اگر کوچک باشد، overhead زیادی خواهیم داشت. این روش برای interactive system مناسب است	Round Robin