زمان‌بندی تک پردازنده ای

Uniprocessor Scheduling

فصل ۹

محمد حسین خسروی – دانشگاه بیرجند

اهداف زمان‌بندی

- تخصیص فرآیندها به پردازنده (ها) در جهت برآورده شدن اهداف سیستم، از قبیل اهداف زیر:
  - زمان پاسخ ، Response time
  - توان عملیاتی، Throughput
  - کارایی پردازنده، Processor efficiency
- زمان‌بندی عبارتست از اداره صفها برای به حداقل رساندن تاخیر صف‌بندی و بهینه سازی کارایی در محیط صف بندی.

انواع زمان‌بندی

- زمان‌بندی بلند مدت:
  - تصمیم گیری برای زمان افزودن به مجموعه فرآیندهای در حال اجرا.
- زمان‌بندی میان مدت:
  - تصمیم گیری در مورد زمان مبادله فرآیندها
- زمان‌بندی کوتاه مدت:
  - تصمیم گیری برای زمان تخصیص پردازنده
- زمان‌بندی ورودی/خروجی:
  - تصمیم گیری برای زمان تخصیص دستگاههای ورودی و خروجی

زمان‌بندی بلند مدت

- مشخص کننده برنامه هایی که برای پردازش در سیستم پذیرفته میشوند.
- درجه چند برنامگی را کنترل میکند.
- سوالات مطرح برای زمانبند:
  - آیا سیستم عامل باید فرآیند اضافی دریافت کند یا خیر؟
    - توسط درجه چندبرنامگی کنترل میشود.
  - کدام کار انتخاب شود؟
    - بطور ساده بر اساس خروج بترتیب ورود.
    - یا بر اساس معیارهایی مانند اولویت، تخمین زمان اجرا و نیازهای ورودی/خروجی

زمان‌بندی میان مدت

- بخشی از عملیات مبادله
- بر اساس نیازهای حافظه ای فرآیندهای مبادله شده به خارج یا میزان نیاز به مدیریت درجه چندبرنامگی.

زمان‌بندی کوتاه مدت

- همان توزیع کننده (dispatcher)
- بیشترین دفعات اجرا را دارد و در مورد انتخاب فرآیند بعدی برای اجرا تصمیم گیری میکند.
- زمانی اجرا میشود که یکی از حوادث زیر اتفاق بیفتد:
  - وقفه های ساعت
  - وقفه های ورودی/خروجی
  - فراخوانیهای سیستم عامل
  - علائم

انواع زمان‌بندی

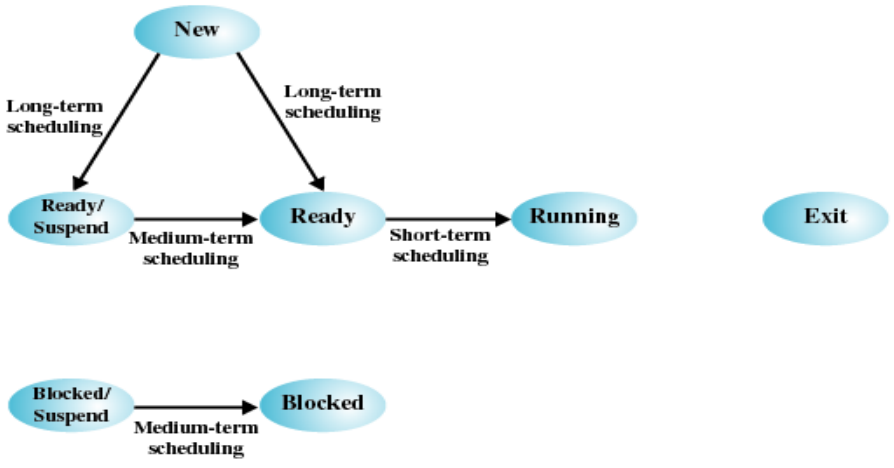


Figure 9.1 Scheduling and Process State Transitions

دسته بندی معیارهای زمان‌بندی

- معیارهای وابسته به کارآمدی (Performance-related)
  - کمّی (Quantitative)
  - قابل اندازه گیری (مانند زمان پاسخ و توان عملیاتی)
- معیارهای غیر وابسته به کارآمدی
  - کیفی (Qualitative)
  - براحتی قابل اندازه گیری و تجزیه و تحلیل نیستند. (مانند قابلیت پیش‌بینی)

معیارهای زمان‌بندی

Table 9.2 Scheduling Criteria

User Oriented, Performance Related

Turnaround time This is the interval of time between the submission of a process and its completion. Includes actual execution time plus time spent waiting for resources, including the processor. This is an appropriate measure for a batch job.

Response time For an interactive process, this is the time from the submission of a request until the response begins to be received. Often a process can begin producing some output to the user while continuing to process the request. Thus, this is a better measure than turnaround time from the user's point of view. The scheduling discipline should attempt to achieve low response time and to maximize the number of interactive users receiving acceptable response time.

Deadlines When process completion deadlines can be specified, the scheduling discipline should subordinate other goals to that of maximizing the percentage of deadlines met.

User Oriented, Other

Predictability A given job should run in about the same amount of time and at about the same cost regardless of the load on the system. A wide variation in response time or turnaround time is distracting to users. It may signal a wide swing in system workloads or the need for system tuning to cure instabilities.

معیارهای زمان‌بندی

System Oriented, Performance Related

Throughput The scheduling policy should attempt to maximize the number of processes completed per unit of time. This is a measure of how much work is being performed. This clearly depends on the average length of a process but is also influenced by the scheduling policy, which may affect utilization.

Processor utilization This is the percentage of time that the processor is busy. For an expensive shared system, this is a significant criterion. In single-user systems and in some other systems, such as real-time systems, this criterion is less important than some of the others.

System Oriented, Other

Fairness In the absence of guidance from the user or other system-supplied guidance, processes should be treated the same, and no process should suffer starvation.

Enforcing priorities When processes are assigned priorities, the scheduling policy should favor higher-priority processes.

Balancing resources The scheduling policy should keep the resources of the system busy. Processes that will underutilize stressed resources should be favored. This criterion also involves medium-term and long-term scheduling.

استفاده از اولویتها

- زمانبند همواره فرآیندی را انتخاب خواهد کرد که دارای اولویت بالاتری باشد.
- برای این منظور میتوان صفهای اولویت متفاوتی در نظر گرفت.

صف بندی بر حسب اولویت

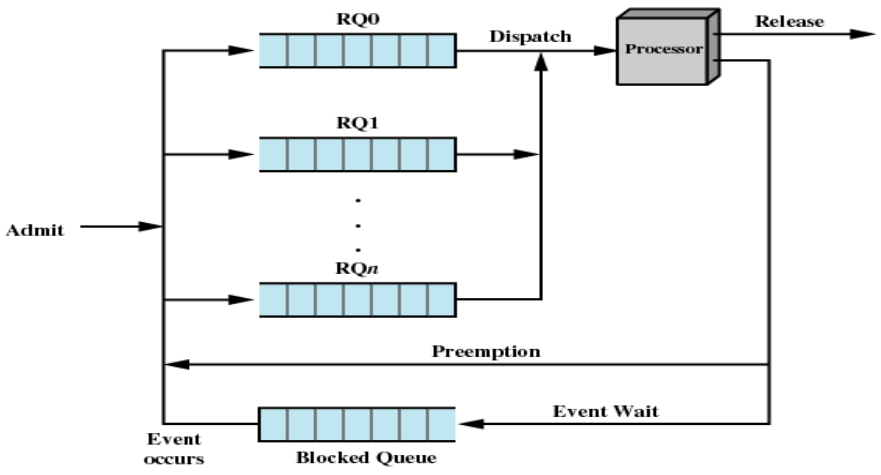


Figure 9.4 Priority Queuing

استفاده از اولویتها

- اشکال این روش اینست که ممکن است فرآیندهای صفهای کم اولویت دچار گرسنگی شوند.

– برای رفع این اشکال میتوان اولویت فرآیند را با افزایش عمر آن یا با تاریخچه اجرای آن تغییر داد.

حالت تصمیم

- مشخص کننده زمانی که تابع انتخاب به اجرا در می‌آید.
- بدون قبضه کردن (Nonpreemptive)

– هنگامی که فرآیندی در حال اجراست، آنقدر به اجرا ادامه می‌دهد تا خاتمه یابد یا خودش برای درخواست I/O مسدود شود.

- با قبضه کردن (Preemptive)

– فرآیند در حال اجرا می‌تواند توسط سیستم عامل متوقف شود و به حالت آماده انتقال یابد.

مثال زمان‌بندی فرآیند

Table 9.4 Process Scheduling Example

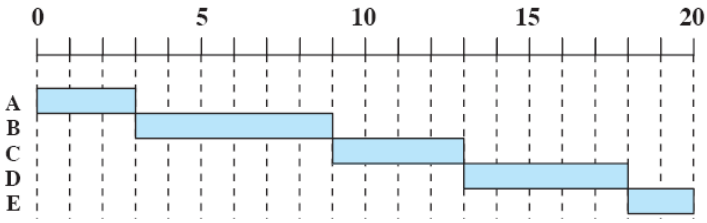
Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

خدمت به ترتیب ورود (FCFS)

First-Come-First Served (FCFS)



- هر فرآیند با آماده شدن به صف آماده ملحق میگردد.
- وقتی فرآیند جاری از اجرا باز میماند، قدیمی ترین فرآیند در صف آماده برای اجرا انتخاب میشود.

18

خدمت به ترتیب ورود (FCFS)

- اشکالات:

– یک فرآیند کوچک ممکن است مدت زمان زیادی را در صف انتظار تخصیص پردازنده معطل بماند.

– مثال:

فرآیند	زمان ورود	زمان خدمت (Ts)	زمان شروع	زمان پایان	زمان کل (Tr)	Tr/Ts
W	0	1	0	1	1	1
X	1	100	1	101	100	1
Y	2	1	101	102	100	100
Z	3	100	102	202	199	1.99
			میانگین		100	26

خدمت به ترتیب ورود (FCFS)

- اشکالات:

–الگوریتم به نفع فرآیندهای در تنگنای پردازنده عمل میکند. (فرآیندهای در تنگنای I/O متضرر خواهند شد.)

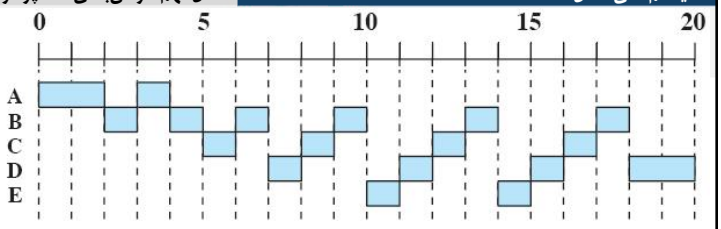
نوبت گردشی Round-Robin

- بکارگیری قبضه کردن بر اساس ساعت
- یک وقفه ساعت در فواصل زمانی دوره ای تولید میگردد.
- با بروز وقفه فرآیندی که در حال اجراست در صف آماده قرار داده میشود و کار آماده بعدی بر اساس FCFS انتخاب میگردد.
- روش برهه زمانی (Time slicing)

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Round-Robin (RR), $q = 1$



زمان	صف
0	**A(3)**
1	**A(2)**
2	**B(6)**, A(1)
3	**A(1)**, B(5)
4	**B(5)**, C(4)
5	**C(4)**, B(4)
6	**B(4)**, D(5), C(3)
7	**D(5)**, C(3), B(3)
8	**C(3)**, B(3), E(2), D(4)
9	**B(3)**, E(2), D(4), C(2)

زمان	صف
10	**E(2)**, D(4), C(2), B(2)
11	**D(4)**, C(2), B(2), E(1)
12	**C(2)**, B(2), E(1), D(3)
13	**B(2)**, E(1), D(3), C(1)
14	**E(1)**, D(3), C(1), B(1)
15	**D(3)**, C(1), B(1)
16	**C(1)**, B(1), D(2)
17	**B(1)**, D(2)
18	**D(2)**
19	**D(1)**

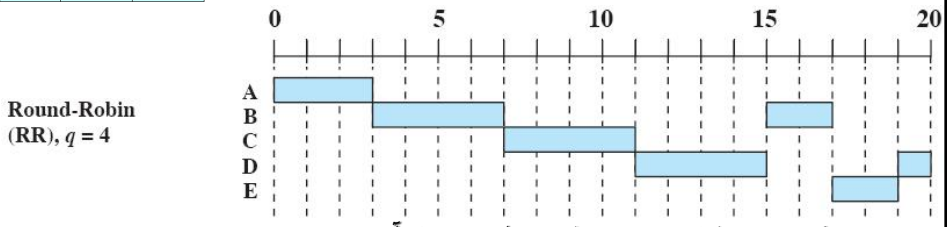
نوبت گردشی Round-Robin

- نکته مهم: اندازه برهه زمانی
  - برهه زمانی کوتاه:
    - سرعت در اجرای فرآیندهای کوتاه
    - سربار پردازش بالا در اداره ساعت و اجرای زمان‌بندی و توزیع فرآیندها
  - برهه زمانی بیشتر از زمان اجرای بلندترین فرآیند:
    - تنزل به FCFS
- بهترین رویکرد:
  - برهه زمانی باید کمی بیشتر از زمان لازم برای یک محاوره متداول باشد.

نوبت گردشی Round-Robin

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



- نوبت گردشی با ۴ برهه زمانی برای هر فرآیند.

زمان	صف
0	**A(3)**
3	**B(6)**
7	**C(4),** D(5), B(2)
11	**D(5)**, B(2), E(2)
15	**B(2)**, E(2), D(1)
17	**E(2)**, D(1)
19	**D(1)**

نوبت گردشی مجازی VRR

- جهت ایجاد عدالت بین فرآیندهای در تنگنای I/O و فرآیندهای در تنگنای CPU.
- فرآیندهای متقاضی I/O پس از انجام درخواست در یک صف کمکی قرار میگیرند.
- فرآیندهای صف کمکی برای توزیع پردازنده دارای اولویت بیشتر هستند.
- وقتی فرآیندی از صف کمکی توزیع میگردد، به اندازه مقدار برهه زمانی منهای زمان آخرین باری که برای اجرا انتخاب شده بود، به اجرا در می آید.

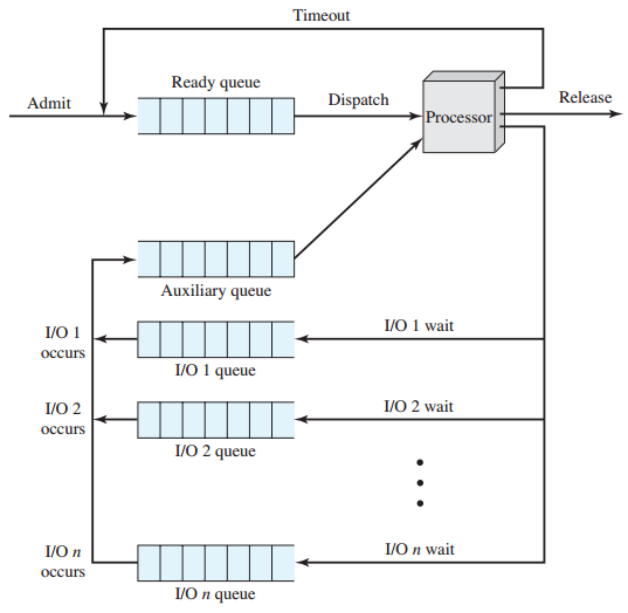


Figure 9.7 Queuing Diagram for Virtual Round-Robin Scheduler

کوتاهترین فرآیند

- Shortest Process Next (SPN)

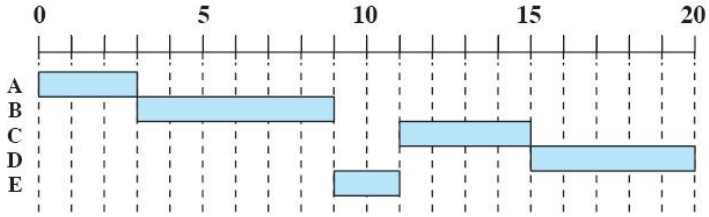
- یک سیاست بدون قبضه کردن.
- فرآیندی که کوتاهترین زمان پردازش مورد انتظار را دارد برای اجرا انتخاب میشود.
- فرآیندهای کوتاه از روی کارهای طولانی گذشته در ابتدای صف قرار میگیرند.
- احتمال گرسنگی فرآیندهای طولانی وجود دارد.

کوتاهترین فرآیند

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Shortest Process Next (SPN)



زمان	صف
0	**A(3)**
3	**B(6)**
9	**E(2),** C(4), D(5)
11	**C(4)**, D(5)
15	**D(5)**

کوتاهترین فرآیند

- مشکل اساسی:
  - نیاز به دانستن زمان پردازش هر فرآیند یا لااقل تخمین آن.
- روشهای حل مشکل:
  - تخمین توسط برنامه ساز.
  - جمع آوری اطلاعات آماری برای محیطهای تولیدی دارای کار مشخص تکراری.
  - محاسبه متوسط دوره های اجرایی برای فرآیندهای محاوره ای

کوتاهترین فرآیند

- یکی از روشهای محاسبه:

مقدار پیش بینی شده برای اجرای $n+1$ ام

$$S_{n+1} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

زمان واقعی اجرای فرآیند برای اجرای i ام

$$S_{n+1} = \frac{1}{n}\left(T_n + \sum_{i=1}^{n-1} T_i\right) = \frac{1}{n}\left(T_n + (n-1)S_n\right)$$

$$= \frac{1}{n}T_n + \frac{(n-1)}{n}S_n$$

کوتاهترین فرآیند

- باز نویسی رابطه مذکور (برای اجتناب از محاسبه کل مجموع در هر بار):

$$S_{n+1} = \frac{1}{n} T_n + \frac{n-1}{n} S_n$$

- **روش متداول، استفاده از متوسط نمایی:**

$$S_{n+1} = \alpha T_n + (1-\alpha) S_n$$

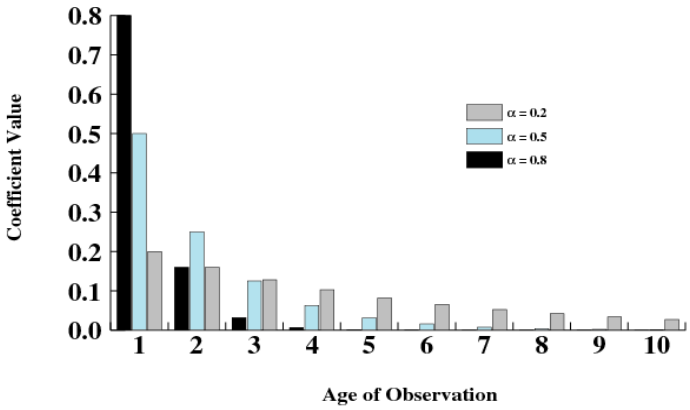
$$= \alpha T_n + (1-\alpha)\alpha T_{n-1} + \cdots + (1-\alpha)^i \alpha T_{n-i} + \cdots + (1-\alpha)^n S_1$$

کوتاهترین فرآیند

- با فرض $\alpha = 0.8$ داریم:

$$S_{n+1} = 0.8\, T_n + 0.16\, T_{n-1} + 0.032\, T_{n-2} + 0.0064\, T_{n-3} + \cdots$$

- ملاحظه میشود که وزن بیشتری به مشاهدات اخیر داده میشود.



کوتاهترین زمان باقیمانده

- Shortest Remaining Time(SRT)

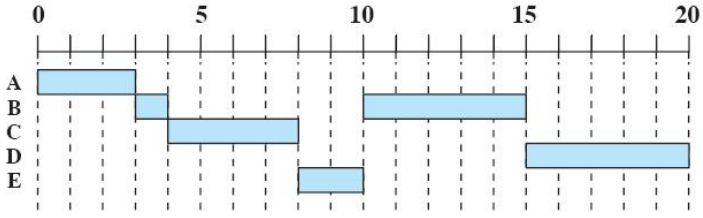
- یک سیاست با قبضه کردن.
- فرآیندی که کوتاهترین زمان پردازش باقیمانده را دارد برای اجرا انتخاب میشود.
- باز هم تخمین زمان پردازش فرآیندها الزامی است.
- احتمال گرسنگی فرآیندهای طولانی نیز وجود دارد.

کوتاهترین زمان باقیمانده

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Shortest Remaining Time (SRT)



زمان	صف
0	**A(3)**
2	**A(1)**, B(6)
3	**B(6)**
4	**C(4)**, B(5)
6	**C(2)**, B(5), D(5)
8	**E(2)**, B(5), D(5)
10	**B(5)**, D(5)
15	**D(5)**

بالاترین نسبت پاسخ

- Highest Response Ratio Next (HRRN)

- در این سیاست فرآیندی که بیشترین مقدار نسبت پاسخ (بر اساس رابطه زیر) را دارد برای اجرا انتخاب میشود.

$$R = \frac{w + s}{s}$$

R: نسبت پاسخ

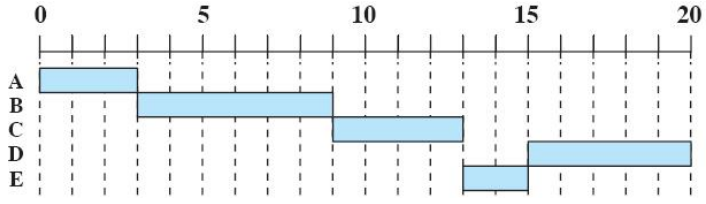
w: زمان انتظار برای پردازنده

s: زمان خدمت مورد انتظار (نیازمند تخمین همانند SPN و SRT)

بالاترین نسبت پاسخ

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Highest Response Ratio Next (HRRN)

$$R = \frac{w + s}{s}$$

زمان	صف
0	A[$R = \frac{0+3}{3} = 1$]
2	B[$R = \frac{1+6}{3} = 2.34$]
9	C[$R = \frac{5+4}{4} = 2.25$], D[$R = \frac{3+5}{5} = 1.6$], E[$R = \frac{1+2}{2} = 1.5$]
13	E[$R = \frac{5+2}{2} = 3.5$], D[$R = \frac{8+5}{5} = 2.6$]
15	D[$R = \frac{10+5}{5} = 3$]

بازخورد (Feedback)

- راهکار اولویت پویا:
  - استفاده از صفهای اولویت در چند سطح.
  - زمان‌بندی بر اساس قبضه کردن(در برهه زمانی)
  - اگر به فرآیندی در صف اولویت RQ_i پردازنده تخصیص یافت اما کارش به اتمام نرسید به صف با اولویت RQ_{i+1} منتقل خواهد شد.
  - در صف دارای کمترین اولویت FCFS و در سایر صفها RR اجرا میشود.

بازخورد (Feedback)

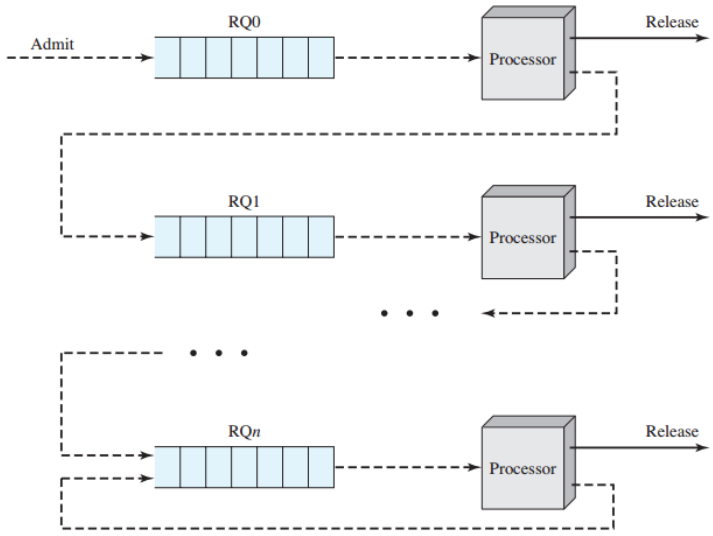
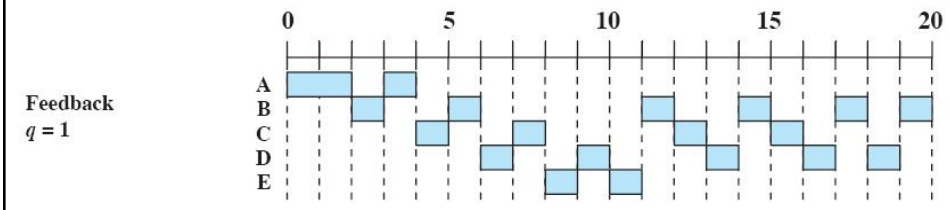


Figure 9.10 Feedback Scheduling

بازخورد (Feedback)

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



- در این سیاست فرآیندی که زمان طولانی تری به اجرا در آمده است مشمول جریمه قرار میگیرد.
- هنگامی استفاده میشود که نتوانیم طول نسبی فرآیندهای متفاوت را مشخص کنیم.

بازخورد (Feedback)

- اشکال:
  - زمان کل فرآیندهای طولانی میتواند بطور نگران کننده ای افزایش یابد.
- راه حل:

1. در نظر گرفتن برهه های زمانی متفاوت برای صفهای کم اولویت.
   - یک فرآیند زمان‌بندی شده برای $\mathbf{RQ_i}$ مجاز است قبل از قبضه شدن، به اندازه $\mathbf{2^i}$ واحد زمانی اجرا شود.
2. ترفیع یک فرآیند به صف با اولویت بالاتر پس از گذراندن مدت زمان مشخصی در صف فعلیش.

بازخورد (Feedback)

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

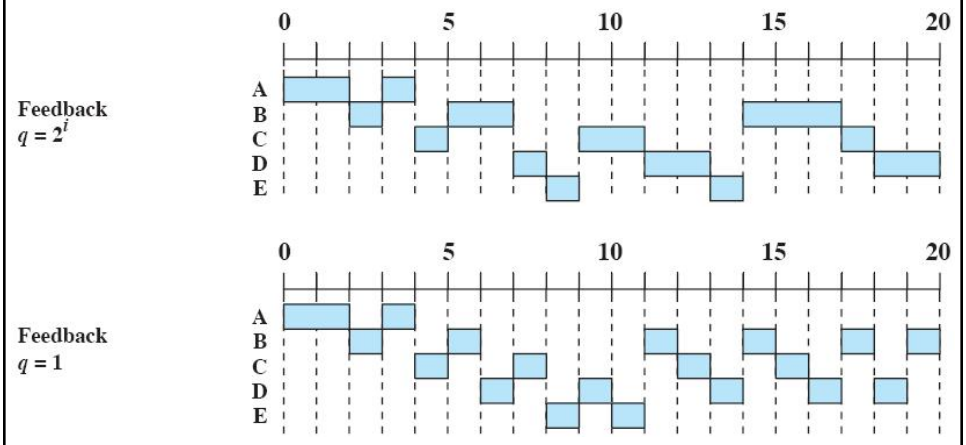


Table 9.3 Characteristics of Various Scheduling Policies

	Selection Function	Decision Mode	Throughput	Response Time	Overhead	Effect on Processes	Starvation
FCFS	max[w]	Nonpreemptive	Not emphasized	May be high, especially if there is a large variance in process execution times	Minimum	Penalizes short processes; penalizes I/O bound processes	No
Round Robin	constant	Preemptive (at time quantum)	May be low if quantum is too small	Provides good response time for short processes	Minimum	Fair treatment	No
SPN	min[s]	Nonpreemptive	High	Provides good response time for short processes	Can be high	Penalizes long processes	Possible
SRT	min[$s - e$]	Preemptive (at arrival)	High	Provides good response time	Can be high	Penalizes long processes	Possible
HRRN	max $\left(\frac{w+s}{s}\right)$	Nonpreemptive	High	Provides good response time	Can be high	Good balance	No
Feedback	(see text)	Preemptive (at time quantum)	Not emphasized	Not emphasized	Can be high	May favor I/O bound processes	Possible

w = time spent waiting
e = time spent in execution so far
s = total service time required by the process, including e

Table 9.5 A Comparison of Scheduling Policies

	Process	A	B	C	D	E	
	Arrival Time	0	2	4	6	8	
	Service Time (T_s)	3	6	4	5	2	Mean
FCFS	Finish Time	3	9	13	18	20	
	Turnaround Time (T_r)	3	7	9	12	12	8.60
	T_r/T_s	1.00	1.17	2.25	2.40	6.00	2.56
RR $q = 1$	Finish Time	4	18	17	20	15	
	Turnaround Time (T_r)	4	16	13	14	7	10.80
	T_r/T_s	1.33	2.67	3.25	2.80	3.50	2.71
RR $q = 4$	Finish Time	3	17	11	20	19	
	Turnaround Time (T_r)	3	15	7	14	11	10.00
	T_r/T_s	1.00	2.5	1.75	2.80	5.50	2.71
SPN	Finish Time	3	9	15	20	11	
	Turnaround Time (T_r)	3	7	11	14	3	7.60
	T_r/T_s	1.00	1.17	2.75	2.80	1.50	1.84
SRT	Finish Time	3	15	8	20	10	
	Turnaround Time (T_r)	3	13	4	14	2	7.20
	T_r/T_s	1.00	2.17	1.00	2.80	1.00	1.59
HRRN	Finish Time	3	9	13	20	15	
	Turnaround Time (T_r)	3	7	9	14	7	8.00
	T_r/T_s	1.00	1.17	2.25	2.80	3.5	2.14
FB $q = 1$	Finish Time	4	20	16	19	11	
	Turnaround Time (T_r)	4	18	12	13	3	10.00
	T_r/T_s	1.33	3.00	3.00	2.60	1.5	2.29
FB $q = 2^i$	Finish Time	4	17	18	20	14	
	Turnaround Time (T_r)	4	15	14	14	6	10.60
	T_r/T_s	1.33	2.50	3.50	2.80	3.00	2.63

زمان‌بندی با سهم عادلانه

- اغلب برنامه کاربر از مجموعه ای از فرآیندها یا نخ ها تشکیل شده است.
- آنچه برای کاربر اهمیت دارد کارایی کل برنامه (مجموعه فرآیندها) است و نه یک فرآیند تکی.
- بنابراین باید تصمیمات زمان‌بندی بر اساس مجموعه فرآیندها باشد.

زمان‌بندی با سهم عادلانه

- روابط زیر برای فرآیند j از گروه k بکار می روند:

$$CPU_j(i) = \frac{CPU_j(i-1)}{2}$$

میزان استفاده فرآیند j از پردازنده در طی فاصله زمانی i

$$GCPU_k(i) = \frac{GCPU_k(i-1)}{2}$$

میزان استفاده گروه k از پردازنده در طی فاصله زمانی i

$$P_j(i) = Base_j + \frac{CPU_j(i)}{2} + \frac{GCPU_k(i)}{4 \times W_k}$$

اولویت پایه فرآیند j

وزن تخصیص یافته به گروه k

اولویت فرآیند j در شروع فاصله زمانی i (عدد کمتر اولویت بالاتر)

زمان‌بندی با سهم عادلانه

Time	Process A Priority	Process A Process CPU count	Process A Group CPU count	Process B Priority	Process B Process CPU count	Process B Group CPU count	Process C Priority	Process C Process CPU count	Process C Group CPU count
0	60	0 1 2 • • 60	0 1 2 • • 60	60	0	0	60	0	0
1	90	30	30	60	0 1 2 • • 60	0 1 2 • • 60	60	0	0 1 2 • • 60
2	74	15 16 17 • • 75	15 16 17 • • 75	90	30	30	75	0	30
3	96	37	37	74	15	15 16 17 • • 75	67	0 1 2 • • 60	15 16 17 • • 75
4	78	18 19 20 • • 78	18 19 20 • • 78	81	7	37	93	30	37
5	98	39	39	70	3	18	76	15	18

Group 1 Group 2

Colored rectangle represents executing process

Figure 9.16 Example of Fair-Share Scheduler—Three Processes, Two Groups

پایان فصل نهم