



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )



دانشکده مهندسی کامپیوتر

به نام خدا

پاسخ تمرین سری نهم درس سیستم های عامل

پاییز 1403

استاد درس: دکتر زرندی

#### توضیحات:

- پاسخ به تمرین ها باید به صورت انفرادی صورت پذیرد. در صورت مشاهده هر گونه تقلب نمره صفر برای کل تمرین منظور خواهد شد.
- تمیزی و خوانایی جواب تمرین ها از اهمیت بالایی برخوردار است. لطفا این مورد را رعایت کنید تا نمره ای به این سبب از شما کسر نگردد.
- لطفا پاسخ تمرین ها را در قالب یک فایل PDF با نام "HW?\_StudentNumber.pdf" در سامانه کورسز و در مهلت معین شده بارگذاری فرمایید.
- در صورت برخوردن به هرگونه مشکل در رابطه با تمرین میتواند از طریق ایمیل [os.fall1403@gmail.com](mailto:os.fall1403@gmail.com) و یا تلگرام با تدریساران در ارتباط باشید.

## سوال اول)

با فرض وجود سه قاب (frame) از الگوریتم های LRU، FIFO و بهینه (optimal) برای رشته های رجوع به صفحه (page) زیر با ذکر مراحل استفاده کرده (از چپ به راست) و در نهایت تعداد نقص صفحه (page fault) را به ازای هر الگوریتم به دست آورید

- ۳, ۷, ۳, ۷, ۶, ۵, ۶, ۳, ۳, ۸, ۷, ۷, ۹, ۵, ۶, ۰, ۲, ۴, ۳, ۵

نقص صفحه ها به رنگ قرمز هستند

### FIFO

3	3	3	3	3	5	5	5	5	5	7	7	7	7	6	6	6	4	4	4
	7	7	7	7	7	7	3	3	3	3	3	9	9	9	0	0	0	3	3
				6	6	6	6	6	8	8	8	8	5	5	5	2	2	2	5

PF = 15

### LRU

3	3	3	3	3	5	5	5	5	8	7	7	7	7	6	6	6	4	4	4
	7	7	7	7	7	7	3	3	3	3	3	9	9	9	0	0	0	3	3
				6	6	6	6	6	6	8	8	8	5	5	5	2	2	2	5

PF = 15

### Optimal

3	3	3	3	3	5	5	5	5	5	7	7	6	6	6	9	9	9	9	9
	7	7	7	7	3	3	3	3	6	6	6	9	9	9	0	2	4	3	3
				6	6	6	6	6	8	5	5	5	5	5	5	5	5	5	5

PF = 11

- 7, 6, 7, 5, 2, 3, 5, 7, 6, 6, 4, 3, 3, 2, 1, 8, 2, 7, 8, 7

FIFO

7	7	7	7	2	2	2	2	6	6	6	6	6	2	2	2	2	7	7	7
	6	6	6	6	3	3	3	3	3	4	4	4	4	0	0	0	0	0	0
			5	5	5	5	7	7	7	7	3	3	3	3	8	8	8	8	8

PF = 13

LRU

7	7	7	7	2	2	2	5	6	6	6	6	6	2	2	2	2	7	7	7
	6	6	6	7	3	3	3	5	5	4	4	4	4	0	0	0	2	2	2
			5	5	5	5	7	7	7	7	3	3	3	3	8	8	8	8	8

PF = 13

Optimal

7	7	7	7	2	3	3	3	6	6	3	3	3	2	2	2	2	2	2	2
	6	6	6	7	7	7	7	3	3	4	4	4	4	0	7	7	7	7	7
			5	5	5	5	5	7	7	7	7	7	7	7	8	8	8	8	8

PF = 10

- 5, 4, 6, 8, 3, 5, 2, 7, 1, 7, 8, 1, 7, 1, 2, 3, 6, 2, 8, 5

FIFO

5	5	5	8	8	8	2	2	2	2	8	8	8	8	8	8	6	6	6	6
	4	4	4	3	3	3	7	7	7	7	7	7	7	2	2	2	2	8	8
		6	6	6	5	5	5	1	1	1	1	1	1	1	3	3	3	3	5

PF = 15

LRU

5	5	5	8	8	8	2	2	2	2	8	8	8	8	7	1	6	6	6	2
	4	4	4	3	3	3	7	7	7	7	7	7	7	2	2	2	2	8	8
		6	6	6	5	5	5	1	1	1	1	1	1	1	3	3	3	2	5

PF = 15

Optimal

5	5	5	8	8	8	2	2	8	8	8	8	8	8	8	8	6	6	6	6
	4	4	5	3	3	3	7	7	7	7	7	7	7	2	2	2	2	2	8
		6	6	5	5	8	8	1	1	1	1	1	1	7	3	8	8	8	5

PF = 12

## سوال دوم

فرض کنید از صفحه آوری مبتنی بر درخواست (demand paging) استفاده می کنیم. جدول صفحات در حافظه اصلی نگه داری می شود که زمان دسترسی به آن ۱۱۰ نانوثانیه است. بنا به ویژگی های حافظه ثانویه در این سیستم، سرویسدهی به نقص صفحه در ۶۵ درصد مواقع ۴ میلی ثانیه و باقی مواقع ۲۱۰ میلی ثانیه طول می کشد. با این مفروضات، بیشترین نرخ نقص صفحه چقدر می تواند باشد تا زمان موثر دسترسی بیشتر از ۲۰۰ نانوثانیه نشود؟

فرض کنید احتمال نقص صفحه  $p$  است

$$t = ((1 - p) * 110) + (p * ((0.65 * 4,000,000) + (0.35 * (210,000,000)))) < 200ns$$

$$76099890p + 110 < 200ns$$

$$p \leq 0.00000118$$

### سوال سوم)

یک حافظه فیزیکی با ۱۰۲۴ قاب، تحت نگاشت یک فضای آدرس دهی منطقی شامل ۲۰۴۸ صفحه که اندازه ی هر صفحه آن ۲ کیلوبایت می باشد ، قرار گرفته است. برای آدرس دهی منطقی و آدرس دهی فیزیکی این فضا به چه تعداد بیت احتیاج داریم؟



طول آدرس منطقی:  $m$   
 طول آفست صفحه:  $n$   
 طول شماره ی صفحات:  $n-m$

$$n = \log_2(\text{size of pages}) = \log_2(2048) = 11$$

$$m - n = \log_2(\text{number of pages}) = \log_2(2048) = 11 \rightarrow m = 22$$

پس به ۲۲ بیت برای آدرس دهی منطقی نیاز داریم

$$m - n = \log_2(\text{number of frames}) = \log_2(1024) = 10, m = 21$$

پس به ۲۱ بیت برای آدرس دهی منطقی نیاز داریم

### سوال چهارم)

با توجه به لیست درخواست شده (از چپ به راست)، ترتیب دسترسی به فضا های خواسته شده را با استفاده از الگوریتم های SSTF (Shortest Seek Time First) و SCAN، C-SCAN، LOOK، C-LOOK بنویسید و همچنین مقادیر Head movement را به ازای هر الگوریتم نیست به دست آورید.  
 - مقدار اولیه سر (head) بر روی ۵۰ است و بازه دیسک از ۰ تا ۱۹۹ است

- ۵۷, ۱۴۰, ۲۳, ۹۸, ۷, ۱۰۲, ۴۸, ۵۲, ۱۷۸, ۱۲

SSTF

48, 52, 57, 23, 17, 12, 7, 98, 102, 140

Head Movement = 194

SCAN

48, 23, 17, 12, 7, 0, 52, 57, 98, 102, 140

Head Movement = 190

C-SCAN

48, 23, 17, 12, 7, 0, 199, 140, 102, 98, 57, 52

Head Movement = 396

LOOK

48, 23, 17, 12, 7, 52, 57, 98, 102, 140

Head Movement = 176

C-LOOK

48, 23, 17, 12, 7, 140, 102, 98, 57, 52

Head Movement = 264

### سوال پنجم)

در چه حالاتی (ترتیبی از درخواست ها) استفاده از الگوریتم C-SCAN بهتر از SCAN است با ذکر مثال دلیل بیاورید. منظور از بهتر بودن لزوماً کمتر بودن Head movement نیست.

الگوریتم C-SCAN در مقایسه با SCAN توزیع عادلانه‌تر زمان انتظار را فراهم می‌کند، زیرا هد تنها در یک جهت درخواست‌ها را سرویس می‌دهد و پس از رسیدن به انتها، مستقیماً به ابتدای مسیر برمی‌گردد. این ویژگی باعث می‌شود درخواست‌های ابتدای مسیر (سیلندرهای پایین) و انتهای مسیر (سیلندرهای بالا) زمان انتظار مشابهی داشته باشند، برخلاف SCAN که در آن درخواست‌های نزدیک ابتدای مسیر باید منتظر برگشت هد باشند.

مثال: هر مثالی که در آن درخواست ها در کل دیسک توزیع شده باشند و مشخص شود که درخواست های در سمت مخالف حرکت اولیه هد دچار انتظار زیادی برای رسیدگی می شوند

