



Department of  
Computer Engineering

# Homework 5-Solution

## Operating Systems

### Fall 2023

Dr. Javadi

## پاسخ سوال یک:

(الف)

با توجه به فرضیات سوال، یک page table در حافظه داریم که دسترسی به آن ۵۰ نانوثانیه زمان میبرد. از طرفی آدرسهای درون page table نیز به خانه هایی در حافظه اشاره میکنند که برای دسترسی به آنها ۵۰ نانوثانیه دیگر زمان لازم است. در نتیجه برای دسترسی به داده یا دستور مورد نظر، به ۱۰۰ نانوثانیه زمان نیاز داریم.

(ب)

با توجه به رابطه زیر داریم:

$$EAT = h \times (\alpha + TLB\_time) + (1 - h) \times (2\alpha + TLB\_time)$$

$$hit\_ratio = 0.75 \text{ and } \alpha = 50ns$$

$$EAT = 0.75 \times (50ns + 2ns) + 0.25 \times (100ns + 2ns) = 64.5ns$$

## پاسخ سوال دو:

(الف) حجمی که می خواهیم در حافظه اصلی قرار بگیرد:

$$0.79 \times 350 = 276.5$$

با توجه به آن که از سیاست اولین مناسب استفاده کردیم، دو سوم حافظه به سبب fragmentation ها از دست می رود، بنابراین حداقل حجم حافظه اصلی مورد نیاز به صورت زیر بدست می آید:

$$276.5 \times 1.5 = 414.75$$

(ب)

17-1 بیت، زیرا 128k معادل  $2^{17}$  است.

19-2 بیت، زیرا 512k معادل  $2^{19}$  است.

3-8، باید سائز آدرس منطقی را بر سائز page ها تقسیم کنیم:

$$128k / 16k = 8$$

4-32، برابر با حاصل تقسیم سایز آدرس فیزیکی بر سایز frame ها خواهد بود و می‌دانیم سایز page ها و frame ها با هم برابر است:

$$512k / 16k = 32$$

5- داریم:

$$\text{Page table size} = \text{number of pages} * \text{size of each page table entry}$$

سابقاً در قسمت 3 تعداد page ها را 8 بدست آوردیم، همچنین هر مدخل page table باید شماره frame مدنظر در حافظه اصلی را نشان دهد، بنابراین برای 32 فریم نیاز به 5 بیت داریم، در نتیجه:

$$\text{PTS} = 8 * 5 = 40 \text{ bits}$$

### پاسخ سوال سه:

آدرس منطقی : 48 بیت

اندازه صفحه: 32 کیلوبایت =  $2^{15}$

ورودی page table: 4 بایت =  $2^2$

فضای آدرس منطقی =  $2^{48}$

اندازه فریم هم با اندازه صفحه برابر است: 32 کیلوبایت =  $2^{15}$

پس داریم:

تعداد صفحه ها:  $2^{48} / 2^{15} = 2^{33}$

اندازه PTS:  $2^{33} * 2^2 = 2^{35} = 32\text{GB}$

اولاً اندازه فریم 32 کیلوبایت است، پس نمیتوانیم page table را در یک فریم قرار دهیم.

پس باید این table رو به father pages تبدیل کنیم و یک page table دیگر بسازیم.

تعداد صفحات در PT1 :

$$\text{PT1 Size / page size} = \frac{2^{35}}{2^{15}} = 2^{20} \text{ B}$$

حالا چک میکنیم که دومین page table در یک فریم جا میشود یا نه:

$$\text{Number of pages at PT2} \times \text{page table entry} = 2^{20} \times 2^2 = 2^{22}$$

پس با توجه به اندازه فریم؛ نمیتوانیم PT2 رو تو یک فریم قرار بدیم.

حالا PT2 رو هم به صفحه ها تقسیم کرده و یک page table دیگر میسازیم. که تعداد ورودی های یکسان دارد.

تعداد صفحات در PT2:

$$\text{Size of PT2 / page size} = \frac{2^{22}}{2^{15}} = 2^7 \text{ B}$$

اندازه PT3:

$$\text{Number of pages at PT3} \times \text{page table entry} = 2^7 \times 2^2 = 2^9$$

پس حالا PT3 در یک فریم جا میشود چون اندازه فریم از اندازه PT3 بزرگتر است.

پس 3 سطح page table نیاز داریم.

### پاسخ سوال چهار:

همانطور که می دانید dirty page به معنای صفحه ای است که زمانی که در حافظه اصلی قرار می گیرد اصلاح می شود و زمانی که می خواهیم صفحه اصلاح شده را جایگزین کنیم به زمان بیشتری برای ذخیره تمام داده های اصلاح شده نیاز دارد.

می دانیم زمانی که page fault رخ میدهد دو حالت پیش می آید :

1. فریم مموری آزاد هست و ما فقط صفحه مورد نیاز رو در حافظه ثانویه کپی می کنیم.
2. حافظه پر شده است در نتیجه باید صفحه را جایگزین کنیم و صفحه خاصی که تصمیم به جایگزینی آن گرفته شده است یک dirty page است.

در این دو حالت دو زمان مختلف برای تکمیل سرویس خطای صفحه مورد نیاز است. معمولاً وقتی صفحه جایگزین اصلاح می شود (کثیف) به زمان بیشتری نیاز دارد زیرا زمان بیشتری برای به روز رسانی بخش اصلاح شده صرف می شود.

نکته : در دو موقعیت زمانی که دو زمان سرویس خطای صفحه مختلف داده می شود، باید میانگین زمان سرویس خطای صفحه را قبل از محاسبه زمان دسترسی موثر حافظه EMAT محاسبه کنیم.

در اینجا، زمان سرویس خطای صفحه = 100 واحد، زمان سرویس خطای صفحه زمانی که صفحه جایگزین شده dirty است = 300 واحد، احتمال dirty بودن صفحه جایگزین شده =  $p$ ، بنابراین، احتمال  $1-p$  بودن صفحه جایگزین شده =  $1-p$

$$\begin{aligned} & \text{Average Page Fault Service Time (pf)} \\ &= \frac{(1-p)(100)}{(\text{No page replacement or not dirty page})} \\ &+ \frac{p(300)}{(\text{Page replacement with dirty page})} \end{aligned}$$

با توجه به این رابطه میانگین زمان سرویس خطای صفحه (Average Page Fault Service Time) برابر است:

$$pfs = 100 - 100p + 300p = 100 + 200p$$

حال باید مقدار EMAT را حساب کنیم:

از آنجایی که  $P$  برابر با احتمال رخ دادن page fault هست و  $pf$  برابر با مقدار کل زمان page fault service هست که در مرحله قبل آن را حساب کردیم  $100 + 200p$  زمان دسترسی به حافظه برابر با یک واحد و زمان دسترسی موثر به حافظه برابر با 3 واحد است.  
در نتیجه :

$$EMAT = p(pfs) + \text{memory access time}$$

$$3 = p(100 + 200p) + 1$$

$$3 = 200p^2 + 100p + 1$$

در نهایت مقدار  $p$  برابر با 0.0194 خواهد بود.

## پاسخ سوال پنج:

الف) تعداد page fault ها و نرخ آن را بدست می آوریم، سپس بر اساس آن زمان موثر دسترسی ها را می یابیم:

FIFO:

0	3	1	4	4	5	2	2	2	2	2	2	0
	0	3	1	1	4	5	5	5	5	5	5	2
		0	3	3	1	4	4	4	4	4	4	5
			0	0	3	1	1	1	1	1	1	4

× × × × × × ×

$$\text{Page fault rate} = \frac{7}{13} = 0.54 \quad \text{EAT} = (1 - 0.54) \times 0.5 + 0.54 \times 5000 = 2700 \text{ms}$$

Optimal:

0	0	0	0	1	1	1	4	5	4	5	5	0
	3	1	1	4	4	4	5	4	5	4	4	5
		3	4	0	5	5	1	1	1	1	1	4
			3	3	0	2	2	2	2	2	2	1

× × × × × × ×

$$\text{Page fault rate} = \frac{7}{13} = 0.54 \quad \text{EAT} = (1 - 0.54) \times 0.5 + 0.54 \times 5000 = 2700 \text{ms}$$

LRU:

0	3	1	4	0	5	2	1	4	5	4	5	0
	0	3	1	4	0	5	2	1	4	5	4	5
		0	3	1	4	0	5	2	1	1	1	4
			0	3	1	4	0	5	2	2	2	1

× × × × × × × ×

$$\text{Page fault rate} = \frac{9}{13} = 0.69 \quad \text{EAT} = (1 - 0.69) \times 0.5 + 0.69 \times 5000 = 3450 \text{ms}$$

رابطه قسمت الف برای بدست آوردن زمان های خواسته شده:

$$\text{EAT} = \text{Page fault rate} \times \text{page fault overhead} + (1 - \text{Page fault rate}) \times \text{memory access time}$$

ب) رشته داده شده را می توان به n قسمت تقسیم کرد که در قسمت i ام بخش زیر را داریم:

1, 2, 3, ..., i

تا زمانی که  $i > r$  نباشد page fault نداریم اما برای زمانی که  $i = r + 1$  شود، در ارتباط با وضعیت frame ها داریم:

r
.
.
.
3
2
1

که برای ارجاع بعدی 1 اخراج می شود و  $r + 1$  در frame ها قرار می گیرد، با ادامه این روند دیده می شود که تمام ارجاعات در ادامه منجر به page fault خواهند شد، بنابراین تعداد کل page fault ها برابر است با:

$$1 + (r + 2) + (r + 3) + \dots + n = ((n - r - 2) + 1) \left( \frac{n + r + 2}{2} \right) + 1$$