

کلاس حل تمرین

مبحث:

Main Memory and Virtual Memory

۱۴۰۲/۱۰/۱۳

روز مادر

یک سیستم ۳۲ بیتی را در نظر بگیرید که از Paged Virtual Memory با اندازه صفحه‌ی 2 KB استفاده می‌کند. برای آدرس مجازی 0x00030f40، هریک از مقادیر Virtual page و Offset مربوط به آن صفحه را بدست آورید.

پاسخ:

برای یک صفحه با اندازه N بایت، تعداد بیت در نظر گرفته شده برای offset برابر با \log_2^N است. بنابراین در این حالت، تعداد بیت offset برابر با ۱۱ بیت است. برای page number هم داریم: $32 - 11 = 21$. پس بطور کلی تعداد 2^{21} صفحه خواهیم داشت که برابر با 2 MPage می باشد.

0000 0000 0000 0011 0000 1	111 0100 0000
31	11 10 0

در اینجا نیز Virtual page number برابر با ۹۷ در مبنای ۱۰ (0x61) و offset برابر ۱۸۵۶ در مبنای ۱۰ (0x740) است.

یک سیستم آدرس دهی ۳۲ بیتی، دارای حافظه اصلی با اندازه‌ی ۴ مگابایت است. اگر اندازه صفحه 1KB باشد، اندازه‌ی Page Table را بدست آورید.

پاسخ:

ابتدا تعداد **entry** مربوط به **page table** را محاسبه میکنیم. برای این کار باید تعداد کل **page** ها را در فضای آدرس منطقی بدست آوریم.

سایز هر صفحه برابر با ۱ کیلوبایت است که میشود 2^{10} بایت، در نتیجه ۱۰ بیت برای **page offset** استفاده میشود و $32 - 10 = 22$ بیت برای شماره صفحه استفاده میشود در نتیجه 2^{22} صفحه خواهیم داشت و به همین تعداد **entry** در **page table** خواهیم داشت.

حال باید مقدار بیت مورد نیاز برای هر **entry** را بدست آوریم. از آنجا در هر **entry** شماره فریم مورد نظر در حافظه اصلی ذکر شده، در نتیجه تعداد فریم های حافظه اصلی مشخص کننده این مقدار خواهد بود. مقدار حافظه اصلی برابر با ۴ مگابایت است که میشود 2^{22} بایت.

سایز هر فریم با سایز هر صفحه یکسان است و برابر است با: 2^{10} بایت. در نتیجه تعداد فریم ها برابر خواهد بود با $2^{12} = \frac{2^{22}}{2^{10}}$. برای نشان دادن این تعداد فریم، به ۱۲ بیت نیاز است و در نتیجه سایز کل **page table** برابر خواهد بود با:

$$2^{22} * 12 \text{ bits}$$

با توجه به ساختار فریم‌ها در شکل زیر، برای هریک از سه الگوریتم اشاره‌شده در شکل، مشخص کنید بعد از درخواست هرکدام از اعضای Page reference string زیر، در هر مورد عضو جدید در کجا قرار می‌گیرد و اینکه آیا fault رخ می‌دهد یا خیر.

Page reference string: A, B, C, A, B, D, A, D, B, C, B

	OPT	FIFO	LRU
ABC	<div></div>	<div></div>	<div></div>
A	<div></div>	<div></div>	<div></div>
B	<div></div>	<div></div>	<div></div>
D	<div></div>	<div></div>	<div></div>
A	<div></div>	<div></div>	<div></div>
D	<div></div>	<div></div>	<div></div>
B	<div></div>	<div></div>	<div></div>
C	<div></div>	<div></div>	<div></div>
B	<div></div>	<div></div>	<div></div>

شکل ۲: Page Replacement Example

پاسخ:

نتیجه نهایی برای هر الگوریتم مطابق شکل زیر است. توجه داشته باشید که اگر همین مفهوم را به نحوه دیگری بیان یا ترسیم کرده باشید، پاسخ شما نیز قابل قبول است.

	OPT	FIFO	LRU
ABC	A B C	A B C	A B C
A			
B			
D	A B D	D B C	A B D
A		D A C	
D			
B		D A B	
C	B	C A B	C B D
B			

فرض کنید در یک حافظه مجازی، مقدار Page reference string ما به صورت زیر است:

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

به ازای هریک از الگوریتم‌های زیر، برای حالت‌های 1, 3, 5, 7 فریم محاسبه کنید چه تعداد page fault رخ می‌دهد. فرض کنید در ابتدا تمام فریم‌ها خالی هستند. یعنی برای هر صفحه برای اولین بار، حتماً یک بار fault رخ می‌دهد.

۱. LRU Replacement

۲. FIFO Replacement

۳. Optimal Replacement

پاسخ:

Number of frames	LRU	FIFO	Optimal
1	20	20	20
3	15	16	11
5	8	10	7
7	7	7	7

(آ) عملیات Swapping را توضیح دهید.

(ب) Page Fault تحت چه شرایطی اتفاق می‌افتد؟ عملکرد سیستم عامل پس از Page Fault به چه صورتی خواهد بود؟

(ج) منظور از Belady's Anamoly چیست؟

(د) درمورد ارتباط بین تعداد فریم‌ها و تعداد Page Fault ها به طور مختصر توضیح دهید.

پاسخ:

الف) خطای صفحه زمانی رخ می‌دهد که درخواست دسترسی به صفحه‌ای که در حافظه اصلی وجود ندارد، رخ دهد. سیستم عامل دسترسی به حافظه را کاملاً کنترل می‌کند و در صورت `invalid` بودن، برنامه را لغو می‌کند. اما در صورتی که `valid` باشد، I/O درخواست می‌شود تا صفحه مورد نیاز را بخواند. فریم‌ها که پس از اتمام فرایند I/O، درواقع جدول فرآیند و Page table ها هستند، بروزرسانی شده و دستورالعمل دوباره راه‌اندازی می‌شود.

ب) فرایندی که به طور موقت برنامه‌های غیرفعال را از حافظه اصلی یک سیستم خارج می‌کند، Swapping نام دارد.

ج) این پدیده در Virtual Memory با Demand Paging رخ می‌دهد و طبق آن افزایش تعداد فریم‌ها منجر به افزایش تعداد Page Fault برای الگوهای خاصی از Access Memory می‌شود. این پدیده هنگام استفاده از الگوریتم FIFO به عنوان الگوریتم Page Replacement بسیار رایج است.

د) عموماً تعداد Page Fault با تعداد فریم‌ها رابطه معکوس دارند. یعنی با افزایش تعداد فریم‌ها، تعداد Page Fault ها، کاهش می‌یابد. البته اگر برای Page Replacement از الگوریتم‌هایی مثل FIFO استفاده کنیم، ممکن است در اثر افزایش تعداد فریم‌ها، Belady's Anomaly رخ دهد و تعداد Page Fault ها، افزایش پیدا کند.