

Amirali Farazmand

99522329

HW5

---

1. به سوالات زیر پاسخ دهید:

الف) آدرس فیزیکی و آدرس منطقی را با هم مقایسه کنید.

ب) در نظر گرفتن page های کوچک و بزرگ هرکدام چه مزیتی میتواند داشته باشد؟

---

الف) آدرس فیزیکی آدرسی است در مموری واقعا وجود دارد، یوزر با physical address به طور مستقیم سروکار ندارد ، (logical address (virtual address آدرسی است که از طرف CPU تولید میشود و طی فرایندی ،توسط (memory-management-unit(MMU به آدرس فیزیکی مپ میشود.

[مقایسه ی کامل تر و جواب کامل به سوال\(GfG\):](#)

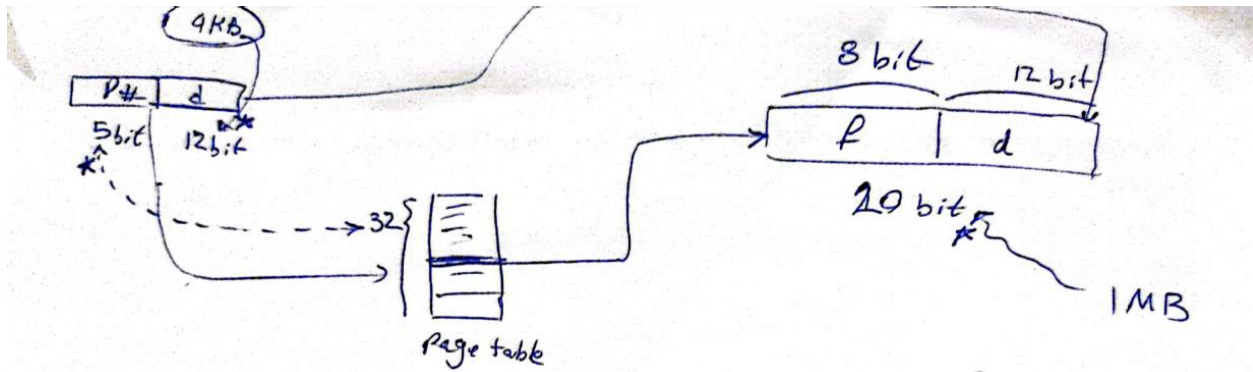
#### Differences Between Logical and Physical Address in Operating System

1. The basic difference between Logical and physical address is that Logical address is generated by CPU in perspective of a program whereas the physical address is a location that exists in the memory unit.
2. Logical Address Space is the set of all logical addresses generated by CPU for a program whereas the set of all physical address mapped to corresponding logical addresses is called Physical Address Space.
3. The logical address does not exist physically in the memory whereas physical address is a location in the memory that can be accessed physically.
4. Identical logical addresses are generated by Compile-time and Load time address binding methods whereas they differs from each other in run-time address binding method. Please refer [this](#) for details.
5. The logical address is generated by the CPU while the program is running whereas the physical address is computed by the Memory Management Unit (MMU).

Parameter	LOGICAL ADDRESS	PHYSICAL ADDRESS
Basic	generated by CPU	location in a memory unit
Address Space	Logical Address Space is set of all logical addresses generated by CPU in reference to a program.	Physical Address is set of all physical addresses mapped to the corresponding logical addresses.
Visibility	User can view the logical address of a program.	User can never view physical address of program.
Generation	generated by the CPU	Computed by MMU
Access	The user can use the logical address to access the physical address.	The user can indirectly access physical address but not directly.
Editable	Logical address can be change.	Physical address will not change.
Also called	virtual address.	real address.

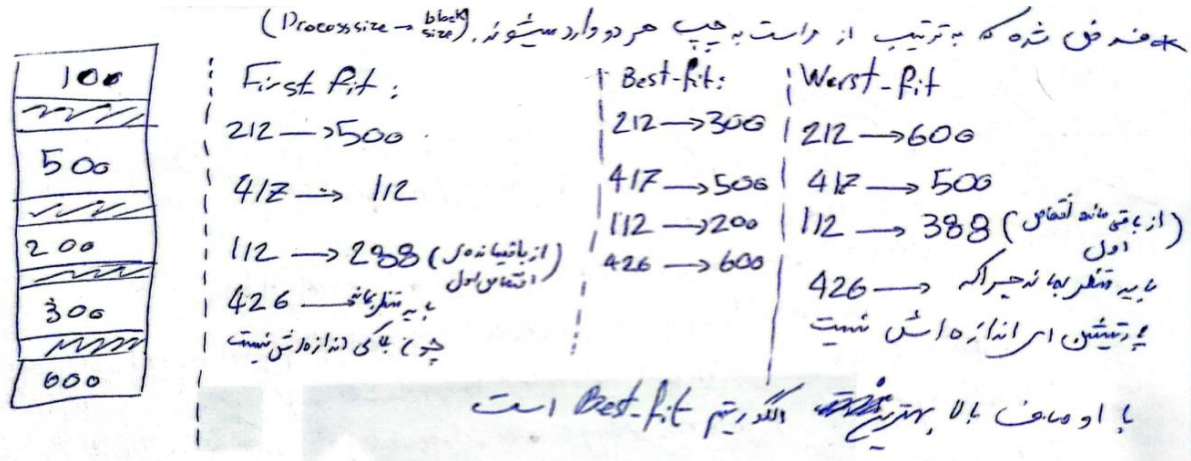
ب) اگر پیچ ما خیلی کوچک باشد، page-table ما خیلی بزرگ میشود و بخش زیادی از مموری را میگیرد. و overhead زیادی از نظر زمان زیادی که در مموری است میدهد. در صورتی که اگر خیلی هم بزرگ باشد فضاهای خالی داخل مموری زیاد میشود و نمیشود بهینه از آن استفاده کرد.

2. در سیستمی که از راهبرد مدیریت حافظه paging پشتیبانی می کند، 32 صفحه 4 کیلوبایتی به یک فضای آدرس دهی 1 مگابایتی نگاشت شده است. اندازه هر درایه جدول باید چند بیتی باشد؟



هر درایه جدول باید 8 بیت داشته باشد  
چون: در شکل اعداد که پیدا شده با \* مشخص شده

3. در سیستمی با پردازنده هایی با اندازه ی 212، 417، 112، 426 و بلوک های حافظه با اندازه های 100، 500، 200، 300، 600 و استفاده از کدام یک از الگوریتم های best fit، worst fit و first fit منطقی تر است؟ راه حل خود را شرح دهید.



همانطور که دیده شد best-fit بهترین الگوریتم است. چرا که همه پراسس ها در مموری قرار گرفتند بعلاوه ی اینکه در این الگوریتم فضاهای خالی کوچکتری پس از allocate بوجود می آیند.

How to satisfy a request of size  $n$  from a list of free holes?

- **First-fit:** Allocate the *first* hole that is big enough
- **Best-fit:** Allocate the *smallest* hole that is big enough; must search entire list, unless ordered by size
  - Produces the smallest leftover hole
- **Worst-fit:** Allocate the *largest* hole; must also search entire list
  - Produces the largest leftover hole

First-fit and best-fit better than worst-fit in terms of speed and storage utilization

4. فرض کنید طرح حافظه demand paging پیاده شده است و جدول صفحه در register ها نگه داری می شود. فرض کنید برای سرویس به یک page fault در صورتی که صفحه ای که باید جابجا شود modify نشده باشد، مدت زمان 8 میلی ثانیه نیاز باشد. این در حالی است که این زمان برای صفحه ای که modify شده است برابر 20 میلی ثانیه است. همچنین زمان دستیابی به حافظه را برابر 100 میکروثانیه در نظر بگیرید. فرض کنید که صفحه ای که باید جابجا شود در 70 درصد مواقع modify می شود. حداکثر نرخ page fault برای حالتی که زمان دستیابی موثر از 200 میکروثانیه بیشتر نشود را محاسبه کنید.

---

Effective Access Time (EAT)

$$\begin{aligned} \text{EAT} = & (1 - p) \times \text{memory access} \\ & + p (\text{page fault overhead} \\ & + \text{swap page out} \\ & + \text{swap page in} ) \end{aligned}$$

$$(1-p) * 100 + p * (0.7 * 20\,000 + 0.3 * 8\,000) \leq 200$$

$$P \leq 0.006134969325$$

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

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

به ازای هریک از الگوریتم های زیر، برای حالت های 3 و 7 فریم محاسبه کنید چه تعداد page fault رخ می دهد. فرض کنید در ابتدا تمام فریم ها خالی هستند. یعنی برای هر صفحه برای اولین بار، حتما یک بار fault رخ می دهد. راه حل خود را به طور کامل بنویسید.

- FIFO Replacement
- LRU Replacement

3 Frames :

• FIFO - Replacement

1	2	5	4	2	1	3	6	2	1	2	5	7	6	5	2	1	2	5	6
*	*	*	*	✓	*	*	*	*	*	✓	*	*	*	✓	*	*	✓	*	*
		<div><div>1</div><div>2</div><div>5</div></div>	4		4	4	6	6	6	<del>5</del>	*	5	<del>5</del>		2	2	2	6	
		5		5	1	1	1	2	2	2	7	7	7	7	1	1	1	1	
					3	3	3	3	1	1	1	6	6	6	5	5	5	5	
→	16	20	0	11															

→ 16 page faults

• LRU - Replacement

1	2	5	4	2	1	3	6	2	1	2	5	7	6	5	2	1	2	5	6
*	*	*	*	✓	*	*	*	*	*	✓	*	*	*	✓	*	*	✓	*	*
1	4	4	3	3	3	1	7	7	7	2	2	2	2	5	5	5	5	5	6
2	2	2	2	6	6	6	5	5	5	5	5	5	5	6	1	6	6	6	6
5	5	1	1	1	2	2	2	2	6	6	6	6	6	6	6	6	6	6	6

→ 15 page faults

7 frames:

• FIFO

1, 2, 5, 4, 2, 1, 3, 6, 2, 1, 2, 5, 7, 6, 5, 2, 1, 2, 5, 6  
 \* \* \* \* \* ✓ ✓ ✓ ✓ \* ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓  
 1  
2  
5  
4  
3  
6  
7

→ 7 page-faults

• LRU

1, 2, 5, 4, 2, 1, 3, 6, 2, 1, 2, 5, 7, 6, 5, 2, 1, 2, 5, 6  
 \* \* \* \* \* ✓ ✓ \* \* ✓ ✓ ✓ ✓ \* ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓  
 1  
2  
5  
4  
3  
6  
7

→ 7 page faults

زیر جاهایی که page-fault میخورد با (\*) مشخص شده و همچنین page-fault های در مراحل اول چون صرفاً فریم های خالی پر میشوند نشان داده نشده اند.

6. در یک سیستم paging، جدول صفحات در حافظه اولیه قرار دارد. زمان دسترسی به حافظه اولیه 50ns و زمان دسترسی به TLB برابر 10ns است. تعداد درایه های TLB برابر 16 می باشد. اگر در ابتدا درایه های TLB همگی خالی بوده و فرآیند در حال اجرا به ترتیب درخواست شماره صفحات آدرس مجازی 2، 3، 1 و 2 را بدهد، درصد افزایش کارایی در هنگام استفاده از TLB نسبت به زمانی که TLB نداشته باشیم، چقدر است؟

---

Without TLB:

Each process needs 50 + 50ns

With TLB:

TLB hits -> 10 + 50ns

TLB miss -> 10 + 50 + 50ns

With TLB we have 3 misses and 1 hit, so:

$$\text{Time} = 3 \times 110 + 1 \times 60 = 390\text{ns}$$

Without TLB we have:

$$\text{Time} = 4 \times 100 = 400\text{ns}$$

$$400/390 = 1.0256$$

افزایش درصد کارایی = 2.56%