سازمان سما

وابسته به دانشگاه آزاد اسلامی

دانشگاه سما واحد حاجی آباد

سیستم عامل

منبع : سیستم عامل دکتر شیر افکن

حمیدرضا رضاپور

WWW.HREZAPOUR.IR

فصل ششم:

مدیریت حافظه

(بخش دوم: حافظه مجازی)

حافظه مجازی

حافظه مجازی تکنیکی است که موجب می شود فرایند بدون اینکه کاملا در حافظه باشد اجرا گردد.

با استفاده از این تکنیک، می توان برنامه ای بزرگتر از حافظه فیزیکی را اجرا کرد.

حافظه مجازی چندبرنامگی را به صورت موثری ممکن می‌سازد و کاربر را از محدودیت های حافظه اصلی رها می‌کند.

به کمک حافظه مجازی دیگر لزومی ندارد تمام صفحه‌ها(یا قطعه‌های) یک فرایند در حال اجرا، در حافظه اصلی قرار داشته باشند و در ابتدا یک یا چند تکه حاوی آغاز برنامه، توسط سیستم عامل به حافظه آورده می‌شود.

حافظه مجازی، حافظه منطقی کاربر را از حافظه فیزیکی تفکیک می کند.

این تفکیک موجب می شود در حالتی که حافظه فیزیکی کم است، حافظه مجازی بزرگی برای برنامه نویس فراهم شود.

حافظه مجازی، برنامه نویسی را ساده می کند، زیرا برنامه نویس نگران حافظه فیزیکی و کد جایگذاری نمی باشد.

اگر در حین اجرا به آدرسی رجوع شود که در مجموعه مقیم در حافظه نباشد، یک وقفه ایجاد می‌شود و این فرایند مسدود می‌شود و تکه مورد نظر وارد حافظه می‌شود، سپس کنترل دوباره به سیستم عامل بر می‌گردد و فرایند مسدود را به حالت آماده برمی گرداند.

به بخشی از فرایند که واقعاً داخل حافظه اصلی قرار دارد، مجموعه مقیم می‌گویند.

عناصر لازم برای موثر بودن حافظه مجازی

۱- حمایت سخت افزاری برای به کارگیری صفحه بندی (یا قطعه بندی)

۲- حمایت نرم افزاری برای انتقال صفحه‌ها (یا قطعه‌ها) بین حافظه ثانوی و حافظه اصلی.

روش های پیاده سازی حافظه مجازی

۱- صفحه بندی درخواستی (Demand Paging)

۲-قطعه بندی درخواستی (Demand Segmentation)

۳-قطعه بندی صفحه بندی شده (Hybrid Paging and Segmentation)

صفحه بندی درخواستی

در صفحه بندی برای حافظه مجازی(صفحه بندی درخواستی)، مانند صفحه بندی ساده، حافظه اصلی به تکه‌های هم اندازه به نام قاب (fram)و برنامه به صفحه‌ها (page)تقسیم می‌شود.

اما بر خلاف صفحه بندی ساده، لزومی ندارد تمام صفحه‌های یک فرایند در قابهای حافظه اصلی باشند تا فرایند اجرا شود و تنها صفحاتی از فرایند که مورد نیاز است به حافظه اصلی آورده می شوند.

این روش تلفیقی از صفحه بندی و مبادله است.

فیلدهای یک درایه جدول صفحه

شماره قاب صفحه (F#)	هدف اصلی از نگاشت صفحه، یافتن شماره قاب صفحه است.
سن (Age)	هر چه کمتر، یعنی آن صفحه مدت بیشتری در حافظه بدون مراجعه بوده است.
بیت مراجعه شده (R) (Referenced)	اگر یک مراجعه، خواندن یا نوشتن به یک صفحه انجام گیرد، این بیت برابر 1 می شود.
بیت تغییر یافته (M) (Modified)	اگر عمل نوشتن بر روی یک صفحه انجام شود، سخت افزار به صورت اتوماتیک این بیت را 1 می کند.
بیت اعتبار (Valid)	اگر این بیت برای صفحه ای 1 باشد، یعنی برنامه نویس از آدرس های درون این صفحه استفاده کرده است. اگر 0 باشد، یعنی صفحه بدون استفاده است.
بیت حضور - غیاب(P/A)	اگر این بیت 1 باشد، یعنی صفحه در حافظه قرار دارد و سایر فیلدهای این درایه قابل استفاده می باشند.
بیت های حفاظت	این بیت ها نوع دسترسی مجاز را مشخص می کند.

خطای نقص صفحه (Page Fault)

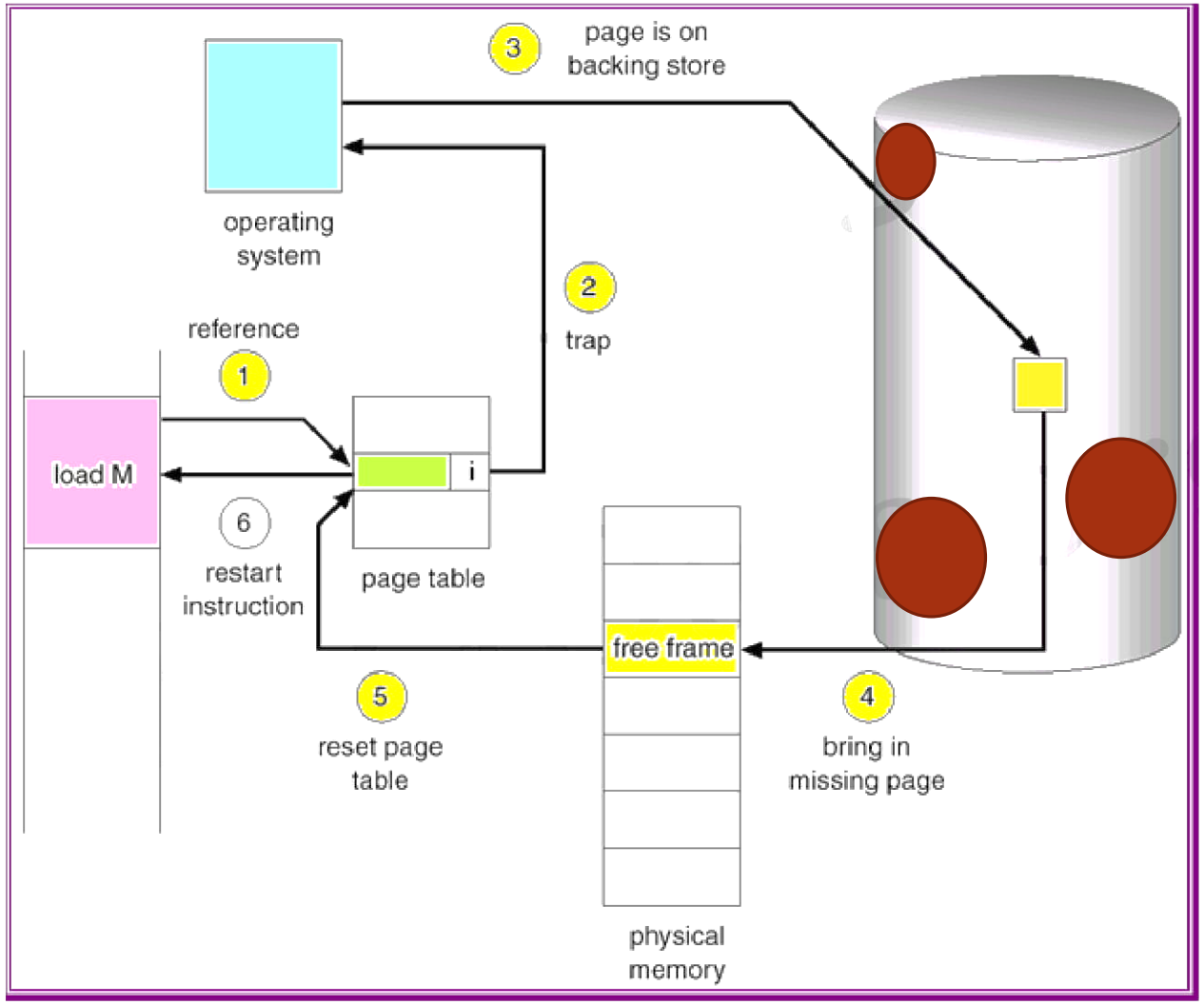
اگر MMU به جدول صفحه مراجعه کند و بیت حضور را برای صفحه مورد نظر صفر ببیند، متوجه می شود که صفحه بر روی دیسک است.

دراین حالت پردازنده در تله (trap) سیستم عامل می افتد.

به این تله، خطای نقص صفحه می گویند.

مدیریت آن بر عهده سیستم عامل است.

Page Fault هندل



مثال

در یک سیستم حافظه صفحه بندی که اندازه هر صفحه برابر ۶۴ بایت می باشد، جدول صفحه زیر را در نظر بگیرید. کدام آدرس خطای صفحه تولید می کند؟

in/out	Frame
out	00101
in	00001
in	11011
in	11010
out	10001
out	10101
out	11000
in	00101
...	...

a) 0000101101001

b) 0000010010010

	in/out
0	out
1	in
2	in
3	in
4	out
5	out
6	out
7	in

a) 0000101 101001 p# = 5 PF

b) 0000010 010010 p# = 2

مثال

یک سیستم حافظه مجازی صفحه بندی را در نظر بگیرید که آدرس مجازی شامل 4096=4k صفحه و هر صفحه شامل 1024=1k بایت است.(آفست ۱۰ بیتی است.)

اندازه حافظه فیزیکی 2^{18} بایت است. آدرس مجازی 111222H از طریق جدول صفحه به آدرس فیزیکی تبدیل می شود. بخشی از جدول صفحه در اینجا مشاهده می شود. آدرس فیزیکی چیست؟

Page	شماره قاب
220H	3CH
221H	3DH
222H	3EH
223H	3FH
224H	40H
⋮	⋮
440H	58H
441H	59H
442H	5AH
443H	5BH
444H	5CH

آدرس مجازی: 111222 H : 00010001000100 1000100010

آدرس فیزیکی : 01 0111 00 1000100010

یعنی : 17222 H

شماره صفحه : 444 H

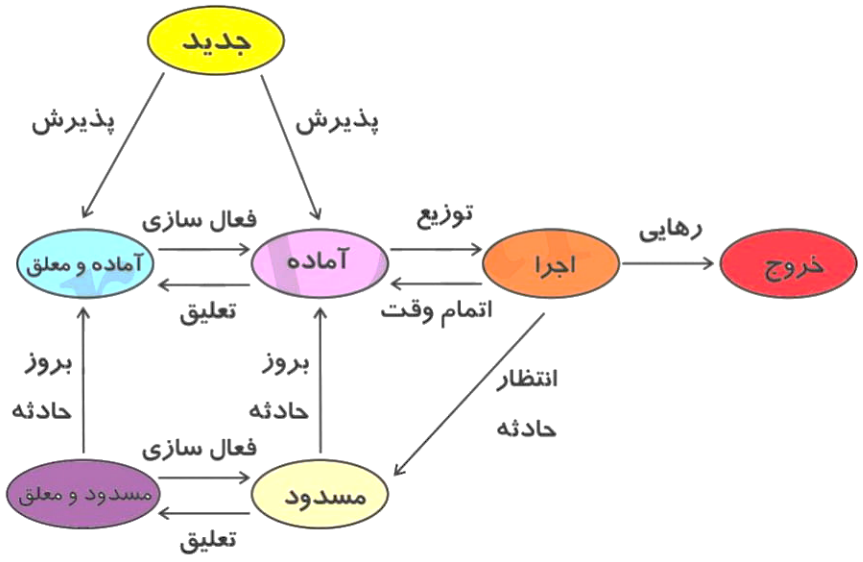
شماره قاب : 5CH (01011100)

الگوریتم‌های جایگزینی صفحه

(Page Replacement Algorithms)

الگوریتم‌های جایگزینی صفحه

در یک سیستم چند برنامه ای که از بخش بندی پویا استفاده می‌کند زمانی خواهد رسید که تمام فرایندهای موجود در حافظه اصلی در حالت مسدود قرار دارند. در این حالت سیستم عامل یکی از فرایندهای داخل حافظه اصلی را به خارج مبادله می‌کند تا فضای کافی برای فرایندی جدید یا یک فرایند آماده و معلق ایجاد شود. سیستم عامل باید فرایندی که قرار است جایگزین شود را انتخاب کند.



در واقع سیاست جایگزینی ، درباره انتخاب یک صفحه برای جایگزینی، در زمانی که لازم است یک صفحه جدید به داخل آورده شود صحبت می‌کند.

الگوریتم های جایگزینی

۱- بهینه (optimal)

۲- عدم استفاده در گذشته اخیر (NRU)

۳- خروج به ترتیب ورود (FIFO)

۴- دومین شانس(Second Change)

۵- ساعت (Clock)

۶- کمترین استفاده در گذشته نزدیک (LRU)

الگوریتم بهینه (optimal)

در این روش، صفحه ای جایگزین می شود که به مدت طولانی مورد استفاده قرار نخواهد گرفت.

الگوریتم بهینه، قابل اجرا نیست، چون نیاز به پیش بینی آینده دارد.

علت مطرح شدن این الگوریتم، استفاده از آن به عنوان شاخص مقایسه می باشد.

نام دیگر این الگوریتم، **BO (Belady Optimal)** می باشد.

مثال

در یک سیستم حافظه صفحه بندی، در یک برنامه به ترتیب به صفحات زیر رجوع شده است. .

برای این برنامه سه قاب صفحه (Page Frame) در نظر گرفته شود.

تعداد خطاهای صفحه (Page Faults) = ؟

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
		1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
F	F	F	F		F		F			F			F				F		

الگوریتم NRU

NRU (Not Recently Used)

در این روش وقتی نقص صفحه رخ می دهد، سیستم عامل بر اساس وضعیت بیت های ارجاع (R) و اصلاح (M) ، صفحات را به چهار کلاس مطابق شکل زیر، تقسیم می کند و سپس به طور تصادفی ، صفحه ای را از کلاسی که در دسته با شماره کمتری باشد، را انتخاب می کند.

شماره کلاس	R	M
0	0	0
1	0	1
2	1	0
3	1	1

در واقع صفحه ای انتخاب می شود که در درجه اول از ابتدای دوره جاری،استفاده نشده است و در درجه دوم، از هنگام ورود به حافظه تغییر نکرده است. صفحه متعلق به دسته شماره 0 ، اخیرا مورد استفاده قرار نگرفته و تغییر نکرده است. به همین علت بهترین انتخاب می باشد.

صفحه ای که بیت اصلاح آن یک باشد، قبل از جایگزینی باید نوشته شود.

کلاس ۱، زمانی رخ می دهد که در کلاس ۳ ، بیت R در یک وقفه ساعت، 0 شود.

شماره کلاس	R	M
0	0	0
1	0	1
2	1	0
3	1	1

وقفه ساعت، بیت M را 0 نمی کند، چون این بیت نشان دهنده این است که آیا این صفحه باید دوباره در دیسک نوشته شود یا خیر.

مثال

حافظه اصلی کامپیوتری دارای چهار قاب صفحه می باشد.

اگر خطای صفحه روی صفحه مجازی شماره ۴ در زمان ۳۱۹ رخ دهد، تحت الگوریتم جایگزینی NRU به ترتیب محتویات کدام یک از قاب صفحه ها ، بایستی جابجا شوند؟

قاب صفحه	شماره صفحه مجازی	زمان بار شدن	زمان آخرین دسترسی	R	M
0	2	125	278	0	1
1	1	229	239	1	0
2	0	119	271	1	0
3	3	159	318	1	1

در روش NRU ، صفحه ۲ انتخاب می شود، چون در دسته با شماره کمتری قرار دارد.

چون در زمان رخ دادن خطای صفحه، یعنی ۳۱۹ ، همه صفحه ها موجود هستند، تصمیم گیری بین همه آنها انجام گرفته است.

الگوریتم FIFO

در این روش(خروج به ترتیب ورود) ، صفحه ای جایگزین شود که زمان بیشتری منتظر بوده است.

برای جایگزینی، صفحه موجود در ابتدای صف را انتخاب کرده و صفحه ای که وارد حافظه شده را به انتهای صف اضافه می کنیم.

مثال

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	0	0	1	2	3	0	4	2	2	2	3	0	0	0	1	2	7
	0	0	1	1	2	3	0	4	2	3	3	3	0	1	1	1	2	7	0
		1	2	2	3	0	4	2	3	0	0	0	1	2	2	2	7	0	1
F	F	F	F		F	F	F	F	F	F			F	F			F	F	F

مثال

0	1	4	2	0	2	6	5	1	2	3	2	1	2	6	2	1	3	6	2
0	0	0	1	4	4	2	0	6	5	1	1	1	1	2	2	3	3	3	6
	1	1	4	2	2	0	6	5	1	2	2	2	2	3	3	6	6	6	1
		4	2	0	0	6	5	1	2	3	3	3	3	6	6	1	1	1	2
F	F	F	F	F		F	F	F	F	F				F		F			F

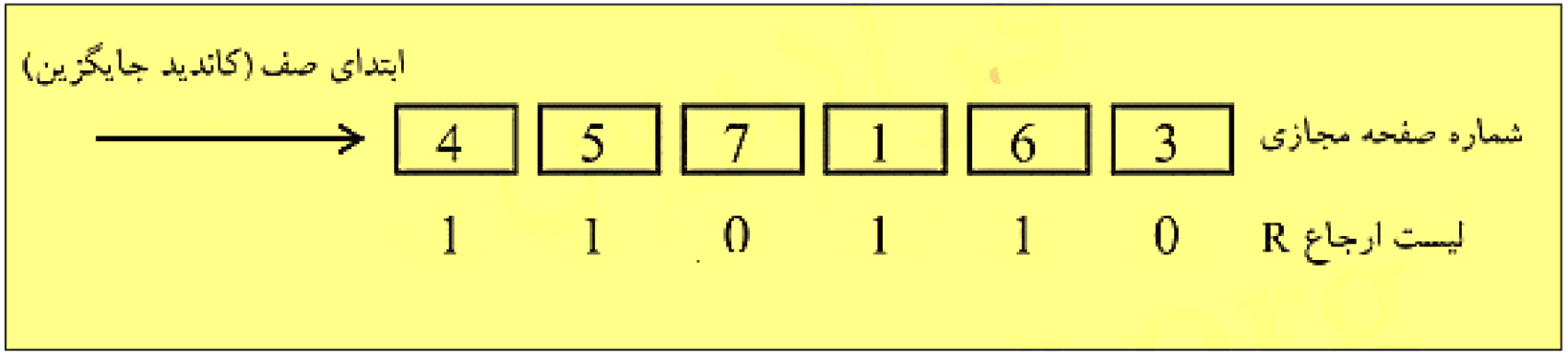
الگوریتم دومین شانس

الگوریتم دومین شانس، مانند FIFO می باشد. با این تفاوت که وقتی نقص صفحه رخ می دهد، اگر بیت R قدیمی ترین صفحه، ۱ بود، این بیت ۰ شده و صفحه به انتهای لیست منتقل می شود. با این کار به او شانس دوباره ای داده شده است.

اگر بیت R همه صفحات ۱ باشد، الگوریتم دومین شانس به FIFO تبدیل می شود.

مثال

ارجاع به شماره صفحه مجازی ۲:



بیت R صفحه های ۴ و۵ چون ۱ است، آن را صفر کرده و صفحه ها به انتهای صف منتقل می شوند.

حال به صفحه ۷ رسیده ایم که چون بیت R آن صفر است، صفحه را از حافظه خارج کرده و صفحه ۲ را به جای آن در انتهای صف اضافه می کنیم.

بیت R صفحه دو را ۱ قرار می دهیم.

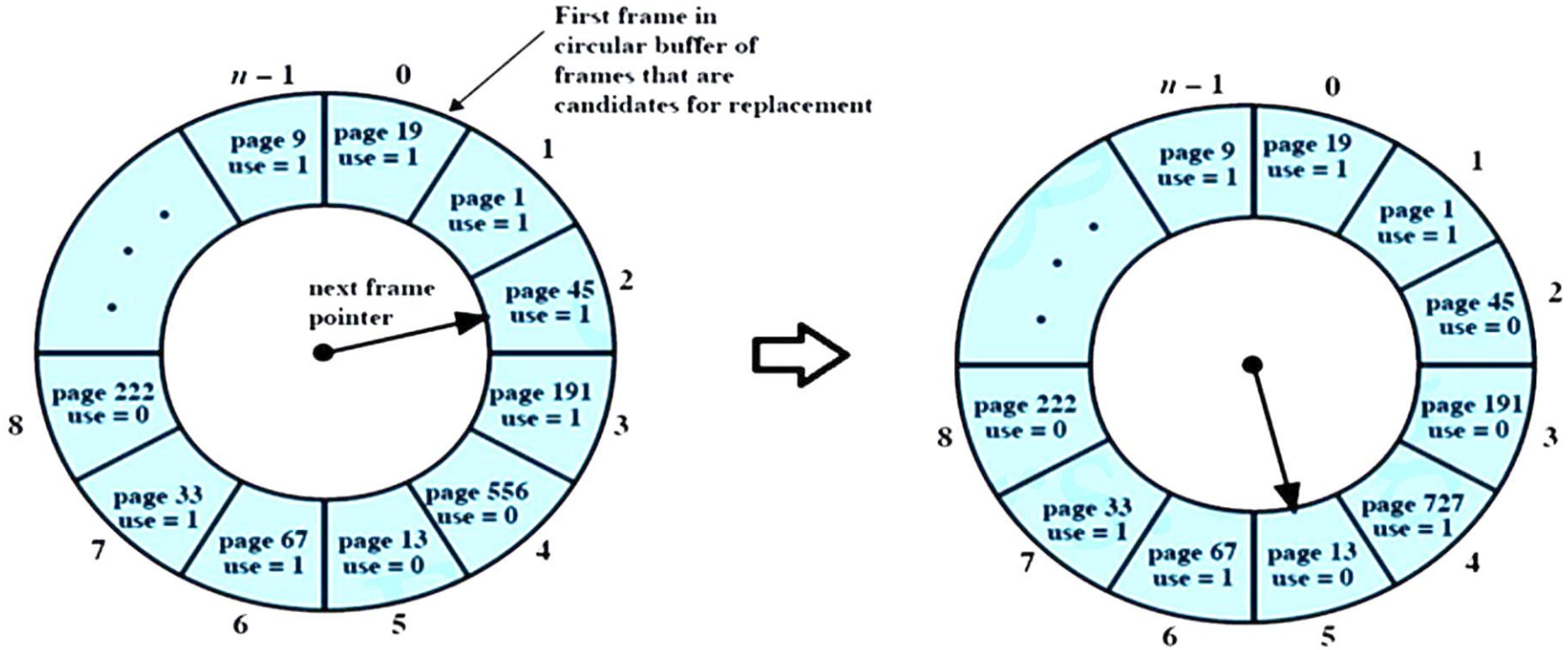
الگوریتم ساعت

اگر چه الگوریتم دومین شانس، الگوریتم قابل قبولی بود، اما به علت حذف صفحه از ابتدای لیست و اضافه کردن آن به انتهای لیست، روشی پر هزینه است. برای حل این مشکل از لیست چرخشی استفاده می شود. در الگوریتم ساعت، لیست پیوندی صفحات به صورت حلقوی و به شکل یک ساعت نگهداری می شود، به طوری که عقربه ساعت به قدیمی ترین صفحه اشاره می کند.

وقتی نقص صفحه رخ می دهد ، به بیت R صفحه ای که توسط عقربه به آن اشاره شده، نگاه کرده که دو حالت رخ می دهد:

الف- اگر بیت R ، 0 باشد، صفحه مورد نظر خارج شده و صفحه جدید در همان مکان جایگزین شده و عقربه به جلو می رود.

ب- اگر بیت R ، 1 باشد، آنگاه 0 شده و عقربه به صفحه بعدی اشاره خواهد کرد.


First frame in circular buffer of frames that are candidates for replacement
next frame pointer
n – 1
0
1
2
3
4
5
6
7
8
page 9 use = 1
page 19 use = 1
page 1 use = 1
page 45 use = 1
page 191 use = 1
page 556 use = 0
page 13 use = 0
page 67 use = 1
page 33 use = 1
page 222 use = 0
n – 1
0
1
2
3
4
5
6
7
8
page 9 use = 1
page 19 use = 1
page 1 use = 1
page 45 use = 0
page 191 use = 0
page 727 use = 1
page 13 use = 0
page 67 use = 1
page 33 use = 1
page 222 use = 0

مثال

سیستمی دارای ۴ قاب صفحه است که مطابق جدول زیر صفحات مجازی در آن ها قرار دارند. اگر سیستم عامل از الگوریتم ساعت (Clock) استفاده نماید، در صورت وقوع نقص صفحه، صفحه جدید با کدام صفحه مجازی باید جایگزین شود؟

شماره صفحه	زمان بارگذاری صفحه درحافظه	R
1	50	1
2	30	0
3	20	1
4	40	0

ابتدا عقربه بر روی قدیمی ترین صفحه یعنی صفحه ۳ است. چون بیت R آن ۱ است، بیت R آن 0 شده و عقربه به جلو حرکت می کند و بر روی صفحه ۲ می رود.

چون بیت R صفحه ۲ ، صفر است، صفحه جدید با این صفحه جایگزین می شود.

الگوریتم LRU

در روش (LRU(Least Recently Used ، صفحه ای جایگزین می شود که برای مدت طولانی مورد استفاده قرار نگرفته است. یعنی در گذشته دورتری به آن مراجعه شده است.

مثال:

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
F	F	F	F		F		F	F	F	F			F		F		F		

مثال

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
1	1	1	1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	5	5	5	5	5	3	3	3	3	3	3	3	3	3
			4	4	4	4	6	6	6	6	6	7	7	7	7	1	1	1	1
F	F	F	F			F	F				F	F	F			F			

مثال

به فرایندی چهار قاب تخصیص یافته است. یک خطای صفحه برای صفحه 4 رخ داده است.

برای سیاست مدیریت حافظه FIFO و LRU به ترتیب چه صفحه ای برای جایگزینی انتخاب می شوند؟

شماره صفحه	شماره قاب	زمان بار شدن صفحه در حافظه	زمان آخرین مراجعه	R	M
2	0	60	160	0	1
1	1	130	161	0	0
0	2	26	163	1	0
3	3	20	162	1	1

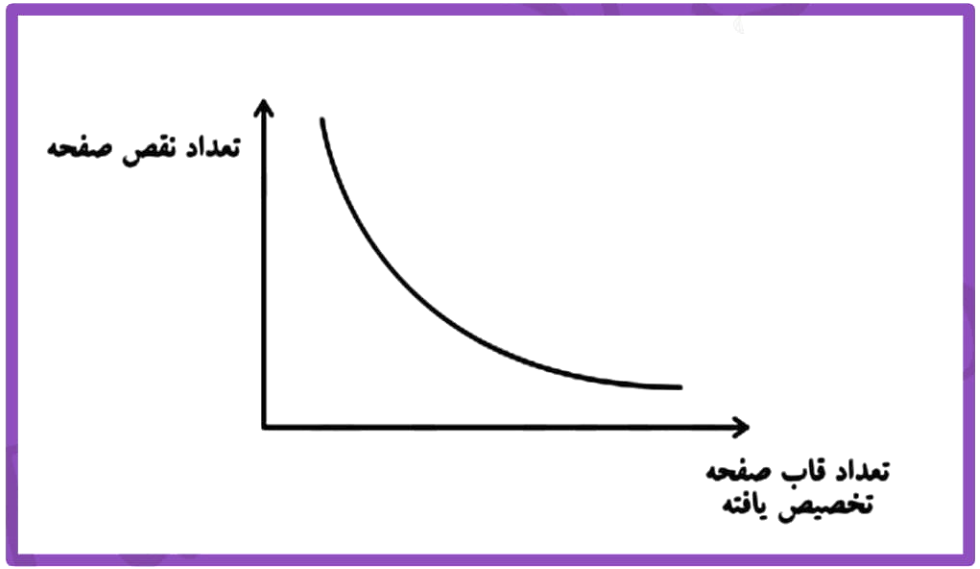
زمان ها بر حسب ضربان ساعت و از شروع فرایند در زمان صفر است.

در روش FIFO صفحه ای که زمان بار شدن آن در حافظه از همه کمتر است انتخاب می شود یعنی صفحه ۳.

در روش LRU صفحه ای که زمان آخرین مراجعه به آن کمتر است، انتخاب می شود یعنی صفحه ۲.

تناقض بلیدی (Belady's anomaly)

در روش FIFO ممکن است با افزایش تعداد قابها، خطای صفحه کم نشود که به این پدیده تناقض بلیدی می گویند. در واقع تناقض بلیدی به این معنی است که الگوریتم FIFO ممکن است از نمودار زیر، پیروی نکند :



تناقض بلیدی در الگوریتم های LRU , BO رخ نمی دهد.

مثال

انباره ۳ صفحه ای : ۹ فقدان صفحه

4	3	2	1	4	3	5	4	3	2	1	5
4	4	4	3	2	1	4	4	4	3	5	5
	3	3	2	1	4	3	3	3	5	2	1
		2	1	4	3	5	5	5	2	1	4
F	F	F	F	F	F	F			F	F	

انباره ۴ صفحه ای : ۱۰ فقدان صفحه

4	3	2	1	4	3	5	4	3	2	1	5
4	4	4	4	4	4	3	2	1	5	4	3
	3	3	3	3	3	2	1	5	4	3	2
		2	2	2	2	1	5	4	3	2	1
			1	1	1	5	4	3	2	1	5
F	F	F	F			F	F	F	F	F	F

تعریف

الگوریتم پشته ای ، الگوریتمی است که در آن، مجموعه ای از صفحات موجود در حافظه برای n قاب، همیشه زیر مجموعه ای از صفحاتی است که برای n+1 قاب در حافظه خواهند بود.

الگوریتم های پشته ای هیچگاه دچار تناقض بلیدی نمی شوند.

قطعه بندی صفحه بندی (Segmentation with paging)

فضای آدرس به تعدادی قطعه تقسیم می‌شود

و هر قطعه نیز به تعدادی صفحه تقسیم می‌شوند.

برای هر فرایند یک جدول قطعه و چند جدول صفحه وجود دارد.

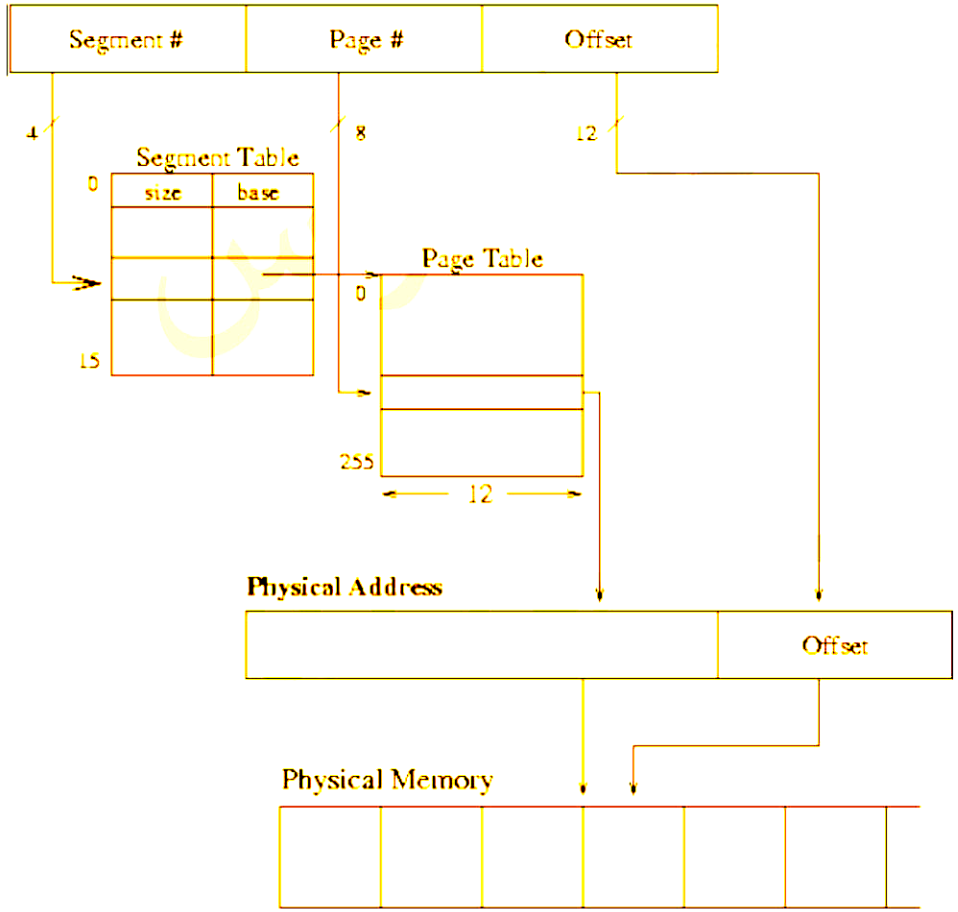
آدرس مجازی :

آفست درون صفحه	شماره صفحه	شماره قطعه

آدرس فیزیکی :

آفست درون صفحه	شماره قاب

قطعه بندی صفحه بندی



مثال

یک حافظه به اندازه 8KB و با صفحات ۵۱۲ بایتی را در نظر بگیرید که با روش قطعات صفحه بندی شده مدیریت می شود.

مطلوب است آدرس فیزیکی متناظر با آدرس مجازی :

S#	P#	Offset
01	010	110101100

F#	offset
1001	110101100

	PTBA	Limit
0	0080^H	8^H
1	0088^H	8^H
2	0090^H	8^H
3	0098^H	8^H

Segment Table

0080^H	0088^H	0090^H	0098^H
3^H	3^H	7^H	5^H
5^H	5^H	5^H	7^H
7^H	9^H	3^H	3^H
9^H	7^H	9^H	9^H
3^H	3^H	7^H	5^H
5^H	5^H	5^H	7^H
7^H	9^H	3^H	3^H
9^H	7^H	9^H	9^H

بخشی از حافظه فیزیکی

چون شماره قطعه ۱ است به اندیس ۱ در جدول قطعه مراجعه کرده و مقدار PTBA یعنی 0088 مشخص می شود.

حال چون در آدرس مجازی، شماره صفحه 2 است، در حافظه، در ستون دوم که با آدرس 0088 شروع می شود به سطر سوم مراجعه کرده و شماره قاب صفحه که برابر $9H$ است ،مشخص می شود.

پایان