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

یکی از وظایف سیستم عامل، مدیریت حافظه است.

مدیریت حافظه اصلی و مدیریت حافظه دیسک بر عهده سیستم عامل است.

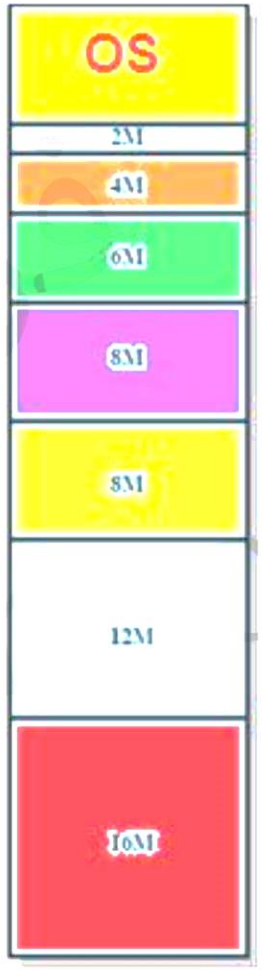
مدیریت حافظه :

۱- تک برنامگی

۲- چند برنامگی

چند برنامگی با پارتیشن ثابت

(a) Equal-size partitions



(b) Unequal-size partitions

معایب مدیریت حافظه به روش پارتیشن بندی ایستا

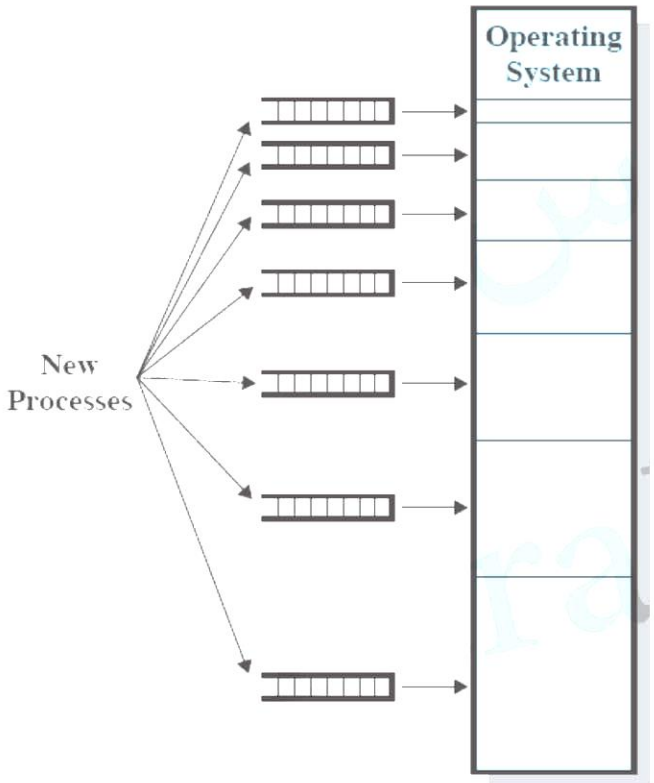
۱- مشکل بودن تعیین تعداد و اندازه پارتیشن ها.

۲- محدود بودن درجه چند برنامگی به تعداد پارتیشن ها .

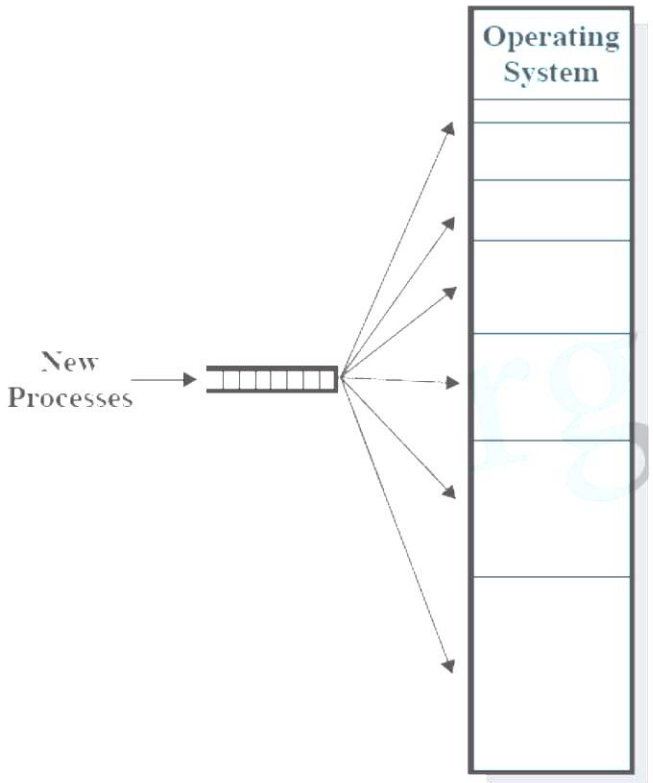
۳- تکه تکه شدن داخلی

صف

هر پارتیشن می تواند، صف مربوط به خود را داشته باشد. می توان یک صف مشترک برای همه پارتیشن ها در نظر گرفت.



(a) One process queue per partition



(b) Single queue

ثبات پایه و حد

هنگامی که پردازنده به فرایندی داده می شود، آدرس شروع پارتیشن در Base و طول پارتیشن در Limit قرار می گیرد.

اگر ثبات پایه حاوی 100 و ثبات حد شامل 20 باشد، آن گاه برنامه می تواند به آدرسهایی از 100 تا خود 120 ، دستیابی داشته باشد.

برای جلوگیری از مشکل جا به جایی، هر آدرس حافظه ای که تولید می شود، قبل از ارسال به حافظه، مقدارش به صورت خودکار با محتوای **Base** جمع می شود.

برای جلوگیری از حفاظت، آدرس ها با محتوای **Limit** مقایسه می شوند تا به فضای خارج از پارتیشن دسترسی نشود.

مبادله (swapping)

در سیستم های اشتراک زمانی در گاهی اوقات نمی توان همه فرایندهای فعال را در حافظه اصلی جای داد و باید فرایندهای اضافی به دیسک منتقل شده و بعدا به صورت پویا به داخل حافظه آورده شوند.

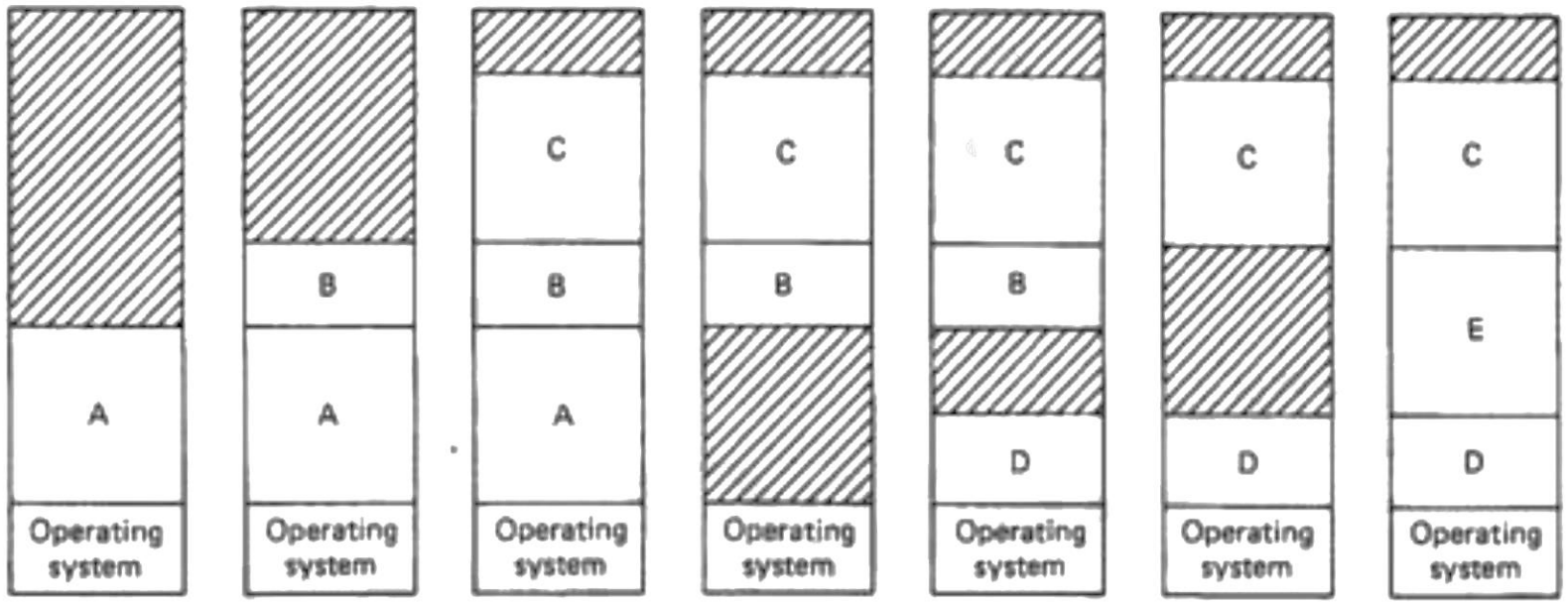
در روش مبادله، هر فرایند به طور کامل به حافظه اصلی آورده می شود و بعد از مدتی اجرا، به دیسک برگردانده می شود .(Swap in ,Swap out)

مثلا در یک محیط چند برنامه ای که از RR استفاده می کند، وقتی فرایندی کوانتوم زمانی اش تمام می شود، با فرایند دیگری مبادله می شود.

در پارتیشن بندی پویا، «تعداد، موقعیت و اندازه» پارتیشن ها متفاوت است. این انعطاف پذیری باعث بهبود بهره وری حافظه می شود.

مثال

شکل زیر نحوه عملکرد سیستمی را نشان می دهد که بر اساس مبادله کار می کند.



اگر انقیاد در زمان اسمبل یا بار کردن باشد، وقتی فرایندی از حافظه بیرون رفت، هنگام برگشت به حافظه، در همان فضای قبلی بر می گردد.

اگر انقیاد در زمان اجرا صورت گیرد، فرایند در محلی غیر از محل اول قرار می گیرد، زیر آدرسهای فیزیکی در زمان اجرا محاسبه می شوند.

تکه تکه شدن خارجی

در پارتیشن بندی پویا، به خاطر مبادله، حفره های متعددی در حافظه به وجود می آید که باعث اتلاف حافظه می شود. به این مشکل، تکه تکه شدن خارجی می گویند.

برای مقابله با تکه تکه شدن خارجی از فشرده سازی می‌توان استفاده کرد.

ساده ترین الگوریتم فشرده سازی این است که تمام حفره ها را به یک طرف برده و حفره بزرگی از حافظه آزاد تشکیل می شود. ولی این روش هزینه زیادی دارد.

تشخیص بخش های آزاد حافظه

وقتی که حافظه به صورت پویا تخصیص داده می شود، سیستم عامل باید بداند که کدام بخش حافظه در هر لحظه، تخصیص داده شده و کدام بخش آزاد است.

برای این منظور، دو روش وجود دارد:

۱- مدیریت حافظه با نگاشت های بیتی

۲- مدیریت حافظه با لیست های پیوندی

مدیریت حافظه با نگاشت های بیتی

حافظه به چندین واحد تخصیص تقسیم می شود.

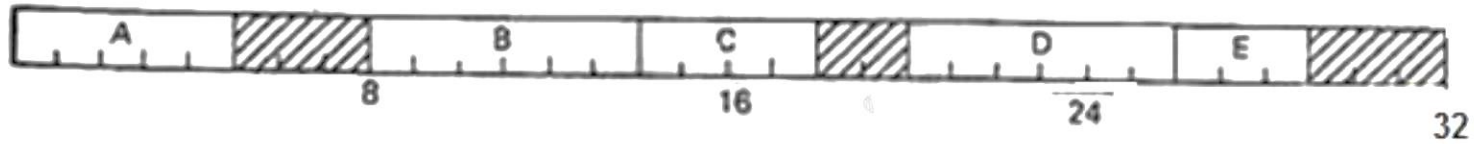
متناظر با هر واحد تخصیص،یک بیت وجود دارد.

اگر واحد متناظر آزاد باشد، این بیت 0 است و در صورت پر بودن، 1 است.

مثال

شکل زیر قسمتی از حافظه شامل 5 فرایند و 3 حفره را نشان می دهد.

مناطق هاشور خورده، فضاهای خالی هستند. به طور مثال، 8 بیت اول ، شامل پنج تا 1 و سه تا 0 است.



11111000
11111111
11001111
11111000

هر چه واحد تخصیص کوچکتر باشد، نگاشت بیتی بزرگتر خواهد بود.

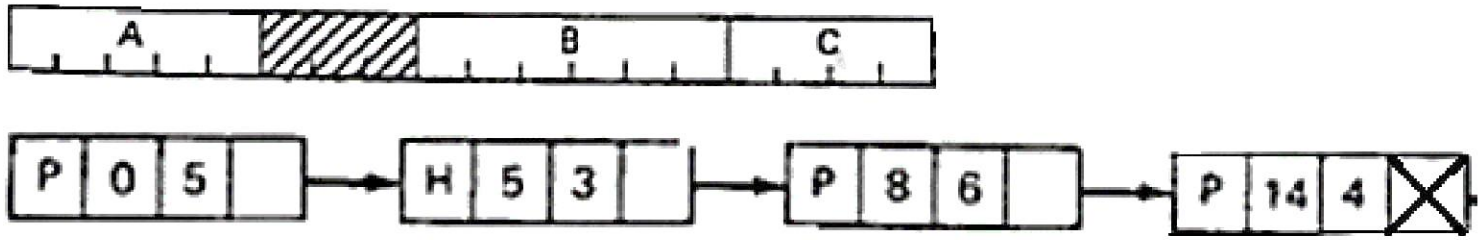
اگر اندازه فرایند، ضریبی از اندازه واحد تخصیص نباشد، در آخرین واحد مقداری از حافظه هدر می رود.

برای انتقال یک فرایند به حافظه که به k واحد فضا نیاز دارد، مدیر حافظه باید در نگاشت بیتی جستجو کرده و k بیت متوالی 0 را پیدا کند. که عملی زمانبر است.

مدیریت حافظه با لیست های پیوندی

یک لیست پیوندی از قطعه های آزاد یا تخصیص یافته حافظه تشکیل می شود.

فضاهای خالی با **H** و فضاهای پر با **P** مشخص شده اند.



فیلدهای گره لیست پیوندی:

۱- حفره یا فرایند بودن ۲- آدرس شروع ۳- طول ۴- آدرس گره بعدی

مزیت این روش این است که زمانی که فرایندی خاتمه می یابد، یا مبادله می شود، این لیست به سادگی **update** می شود.

الگوریتم های مکان یابی و تخصیص حافظه

وقتی فرایندها و حفره ها در یک لیست مرتب شده بر اساس آدرس قرار می گیرند، الگوریتم های مختلفی جهت تخصیص حافظه به یک فرایند وجود دارد.

۱- اولین برازش (First fit)

جستجو از ابتدای حافظه شروع شده و فرایند در اولین حفره ای قرار داده می شود که در آن جا می شود.

۲- برازش بعدی (Next fit)

مانند First fit است، با این تفاوت که جستجو از آخرین محل تخصیص شروع می شود.

۳- بهترین برازش (Best fit)

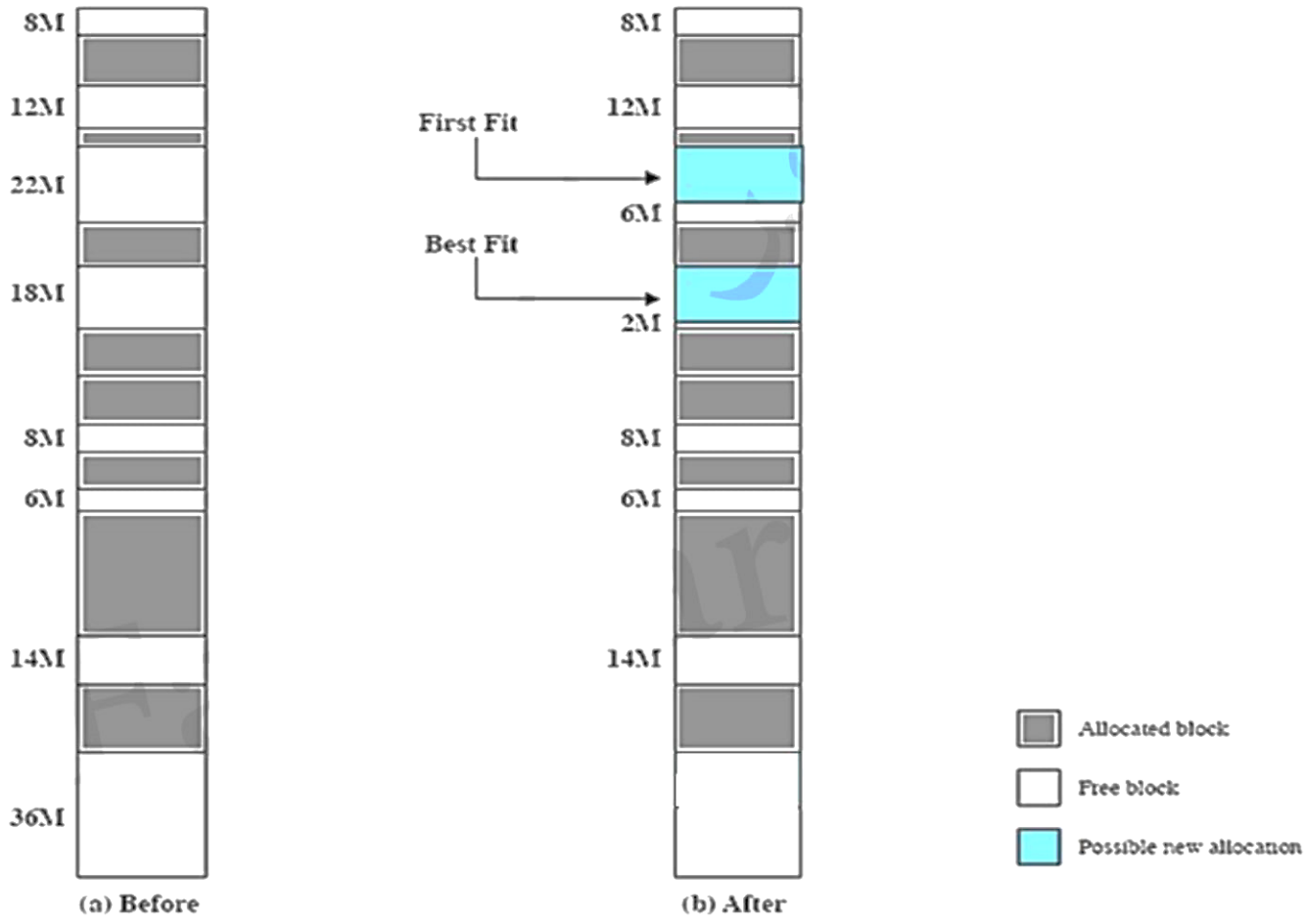
تمام لیست جستجو می شود و فرایند در کوچکترین حفره ای قرار داده می شود که در آن جا می شود.

۴- بدترین برازش (Worst fit)

تمام لیست جستجو می شود و فرایند در بزرگترین حفره ای قرار داده می شود که در آن جا می شود.

مثال

تخصیص فضا به فرایند ۱۶ مگابایتی



مثال

در یک سیستم که مدیریت حافظه با استفاده از مبادله انجام می‌شود، حافظه اصلی شامل فضاهای خالی با اندازه‌های به ترتیب 12 , 9 , 7 , 18 , 20 , 4 , 10(چپ به راست) است. برای درخواست تکه هایی از حافظه به طور متوالی و به مقدارهای 12 و10 و 6 کیلو بایت با استفاده از روش های گفته شده کدام یک از فضاهای خالی فوق الذکر اشغال می‌شوند؟

	10	4	20	18	7	9	12
12	10	4	8	18	7	9	12
10	0	4	8	18	7	9	12
6	0	4	2	18	7	9	12

First fit

Next fit

	10	4	20	18	7	9	12
12	10	4	8	18	7	9	12
10	10	4	8	8	7	9	12
6	10	4	8	2	7	9	12

Best fit

	10	4	20	18	7	9	12
12	10	4	20	18	7	9	0
10	0	4	20	18	7	9	0
6	0	4	20	18	1	9	0

Worst fit

	10	4	20	18	7	9	12
12	10	4	8	18	7	9	12
10	10	4	8	8	7	9	12
6	10	4	8	8	7	9	6

- ✓ اگر لیست فضاهای خالی بر اساس اندازه فضاها مرتب باشد، سرعت **Best fit**، افزایش می یابد.

- ✓ سرعت الگوریتم های **Best fit** و **Worst fit** پایین است، چون کل لیست باید جستجو شود.

- ✓ در **Next fit**، حفره های بزرگ انتهای حافظه سریع تر شکسته می شود و در ورود فرایندهای بزرگ بعدی، مشکل ایجاد می شود.

برازش سریع (Quick fit)

برای هر دسته از فرایندها با اندازه های متداول، یک لیست جداگانه تهیه می شود.

جدولی با n خانه را فرض کنید که :

خانه اول : اشاره گری به ابتدای لیست حفره های 4 کیلو بایتی،

خانه دوم: به ابتدای حفره های 8 کیلو بایتی

و ...

یافتن حفره ای با اندازه مناسب، بسیار سریع است.

مدیریت حافظه با سیستم رفاقتی (Buddy System)

در بخش بندی ایستا امکان تکه تکه شدن داخلی و در بخش بندی پویا ، امکان تکه تکه شدن خارجی وجود دارد.

سیستم رفاقتی، یک تعادل قابل قبول برای فائق آمدن بر معایب طرحهای بخش بندی ایستا و پویا است.

اندازه بلاکهای حافظه توانی از 2 می‌باشند.

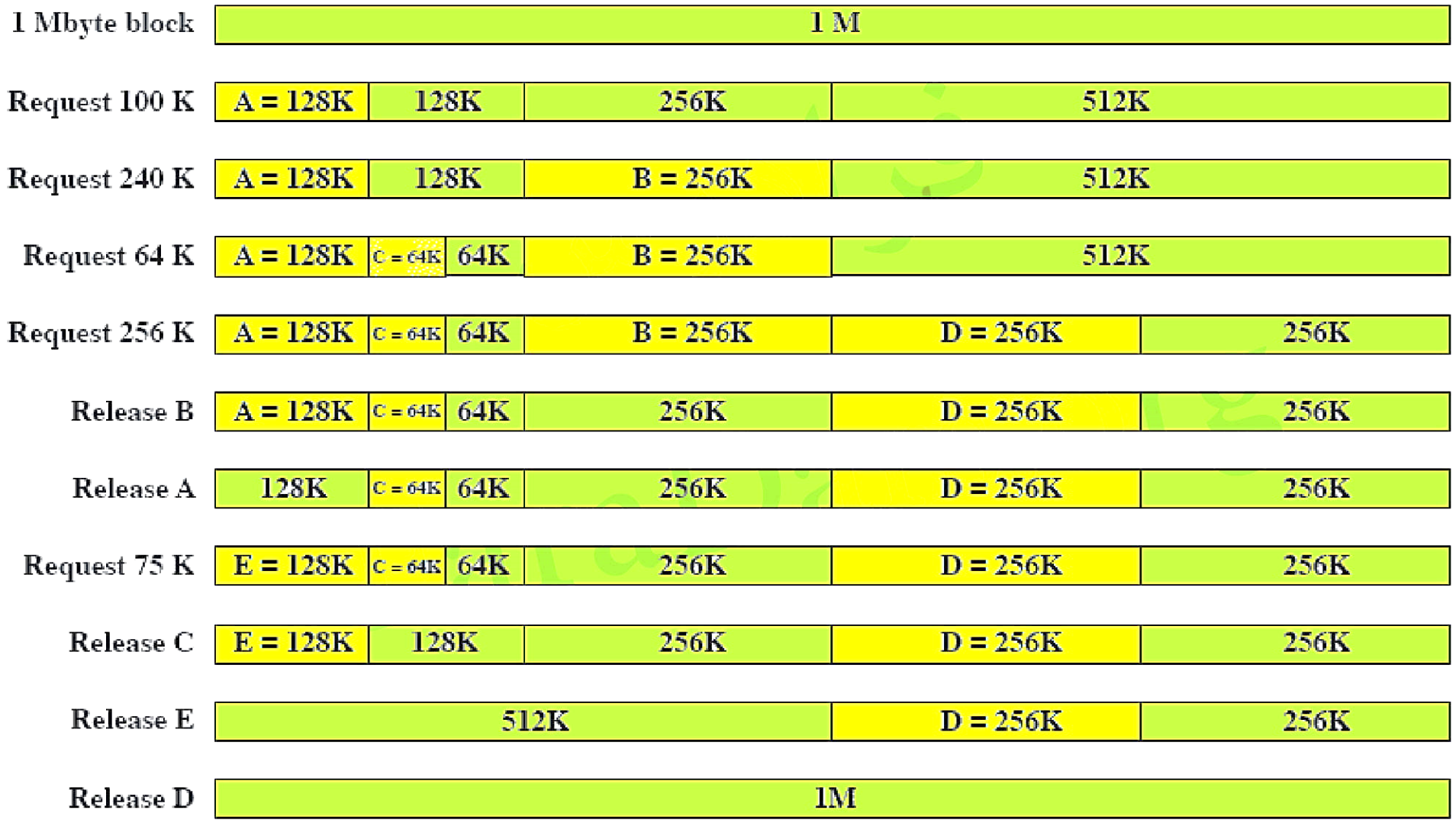
نحوه کار:

اگر اندازه یک حفره برابر 2^k باشد و فرایندی به اندازه S باید به داخل آن مبادله شود، اگر S از نصف اندازه حفره بزرگتر بود، کل فضا به آن داده می شود، در غیر اینصورت،کل بلوک نصف شده و دو بلوک رفیق ایجاد می شود. این روند به صورت بازگشتی تکرار خواهد شد. در صورت آزاد شدن یک حفره، امکان ترکیب رفقای مجاور می باشد.

از معایب سیستم رفاقتی، اتلاف حافظه در این روش است.

مثال

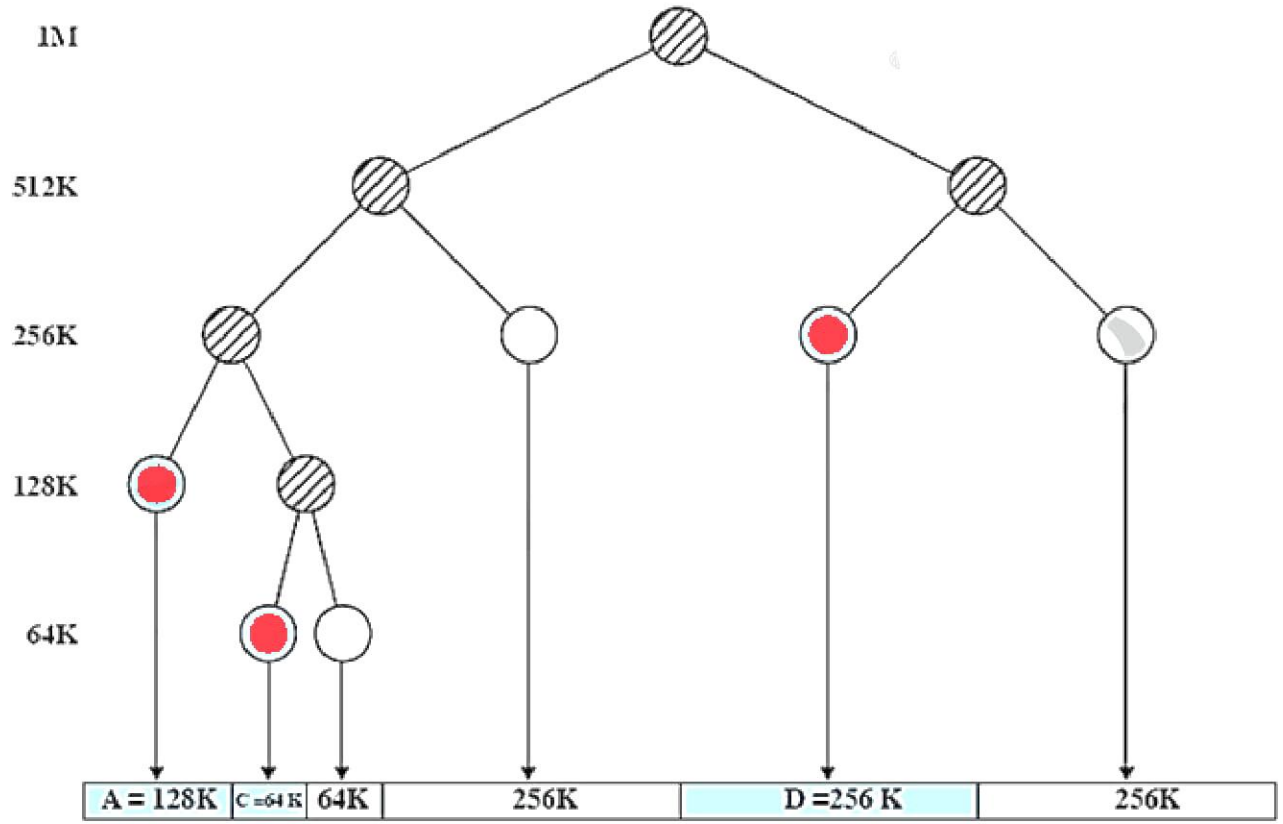
شکل زیر نحوه عملکرد سیستم رفاقتی را نشان می دهد:



ساختار درختی سیستم رفاقتی

مزیت سیستم رفاقتی این است که چیدمان حفره ها ساخت یافته است و می توان یک ساختار درختی برای بلوک های پر و خالی ایجاد کرد و با یک الگوریتم بازگشتی، حفره مناسب را خیلی سریع پیدا کرد.

مثال: درخت دودویی بعد از درخواست **Release B**



صفحه بندی

صفحه بندی (Paging)

حافظه اصلی به بلوکهایی با اندازه های ثابت به نام قاب (frame)تقسیم می شود.

حافظه منطقی به بلوک هایی با اندازه های یکسان به نام صفحه (page)تقسیم می‌شود.

اندازه صفحه و اندازه قاب توسط سخت افزار تعریف می شود.

وقتی یک فرایند به داخل حافظه آورده می‌شود، تمام صفحات آن به داخل قابهای موجود بار می‌شوند و یک جدول صفحه ایجاد می‌شود.

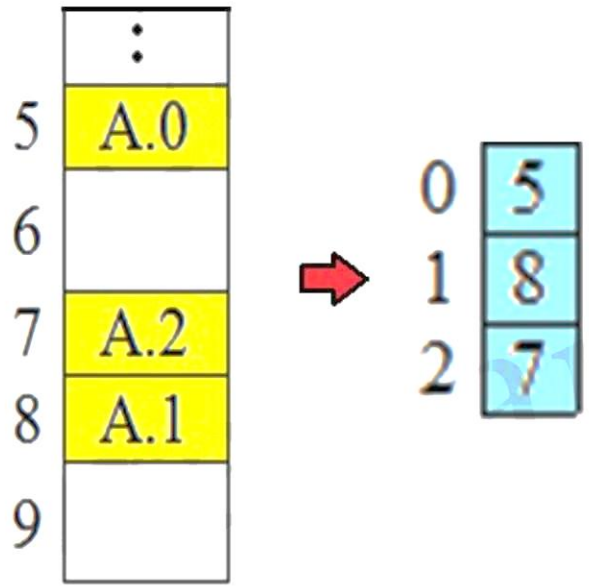
جدول صفحه، محل قاب هر صفحه از فرایند را مشخص می‌کند.

اندازه صفحه و اندازه قاب باید توانی از ۲ باشند.

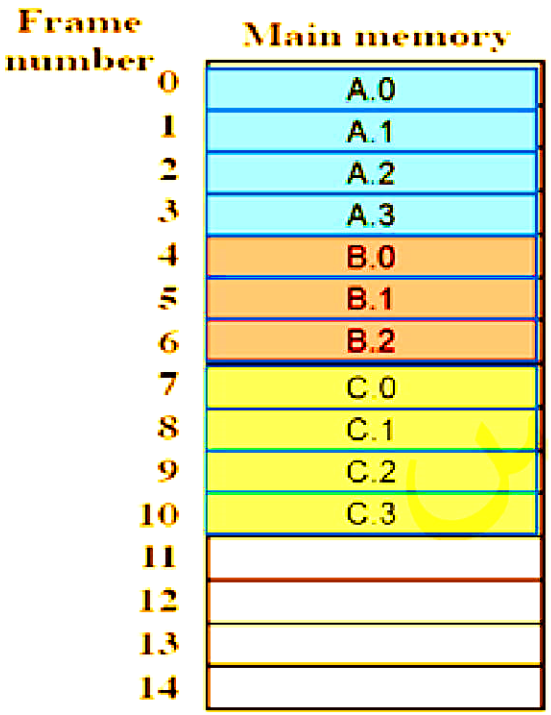
نقطه ضعف اصلی صفحه بندی تکه تکه شدن داخلی است.

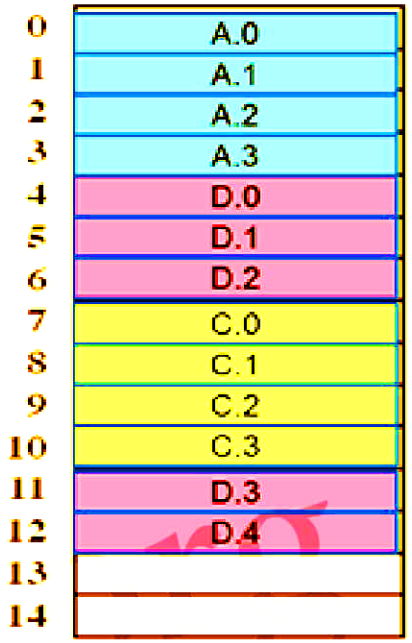
مثال

تخصیص قاب های آزاد به صفحه‌های فرایند A و نمایش جدول صفحه :



مثال





0	0
1	1
2	2
3	3

Process A page table

0	—
1	—
2	—

Process B page table

0	7
1	8
2	9
3	10

Process C page table

0	4
1	5
2	6
3	11
4	12

Process D page table

13
14

Free frame list

ترجمه آدرس در سیستم صفحه بندی

آدرس منطقی از دو قسمت تشکیل شده است:

۱- شماره صفحه (P) ۲- آفست (d)

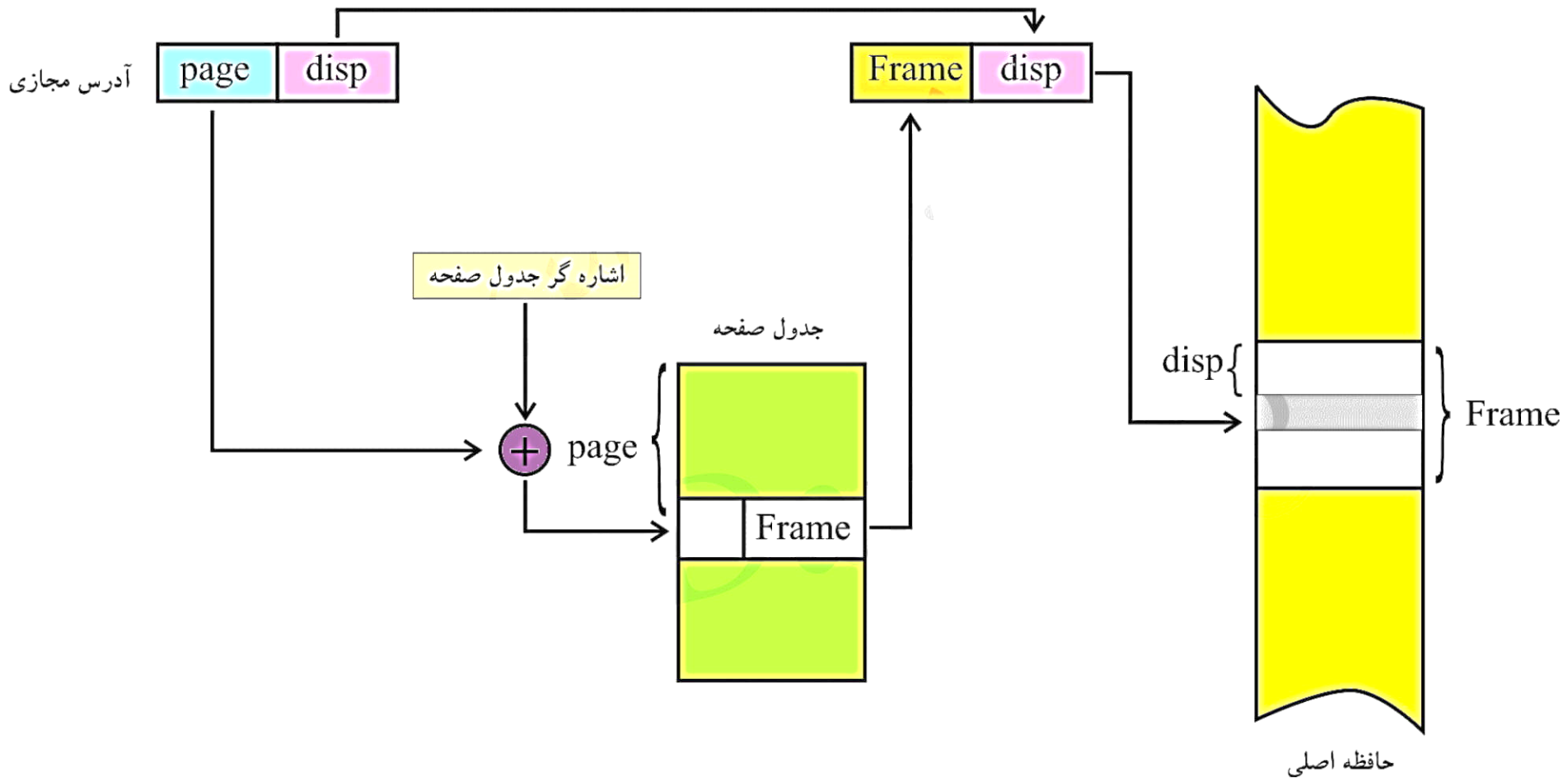
پردازنده به کمک جدول صفحه، یک آدرس فیزیکی تولید می‌کند.

آدرس فیزیکی از دو قسمت تشکیل شده است:

۱- شماره قاب ۲- آفست

به آفست، انحراف یا تفاوت مکان نیز می گویند.

نحوه ترجمه آدرس در سیستم صفحه بندی



مثال

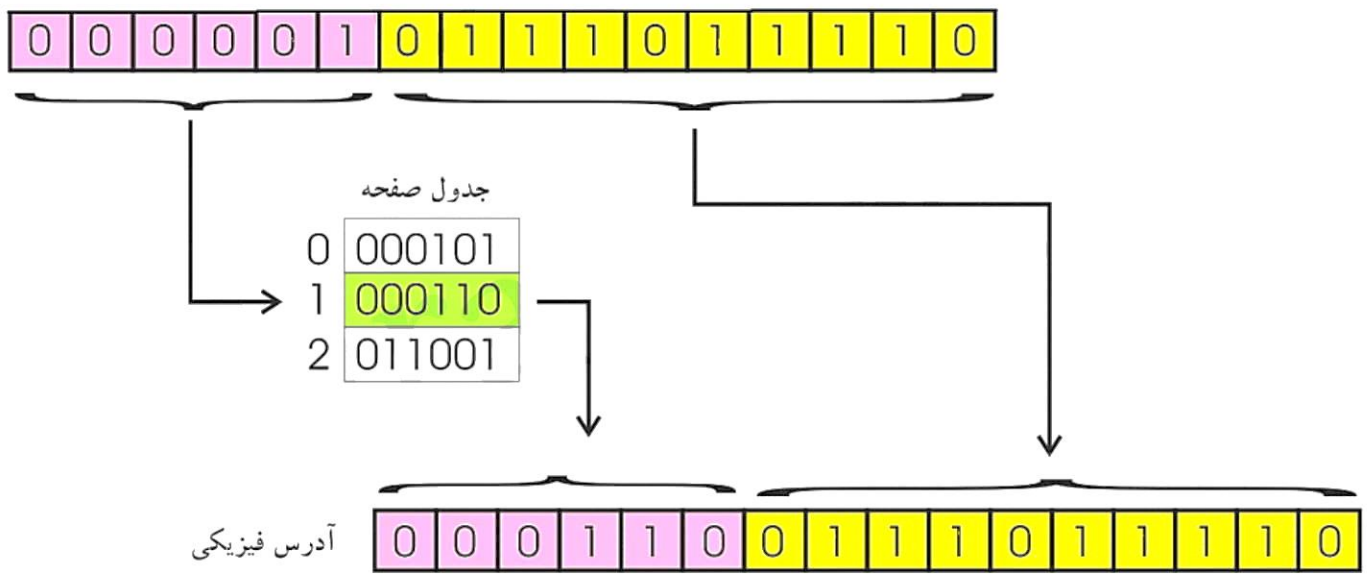
با توجه به آدرس نسبی ۰۰۰۰۰۱۰۱۱۱۰۱۱۱۱۰ ، چند بیت برای شماره صفحه نیاز است؟

(اندازه صفحه برابر یک کیلو بایت می باشد.)

پاسخ: اندازه صفحه یک کیلو بایت است، پس۱۰ بیت برای آفست مورد نیاز است.

چون آدرس ۱۶ بیتی است، ۶ بیت برای شماره صفحه باقی می‌ماند.

آدرس نسبی داده شده دارای آفست (۰۱۱۱۰۱۱۱۱۰) در صفحه (۰۰۰۰۰۱) است.



یک برنامه می‌تواند حداکثر $2^6 = 64$ صفحه یک کیلوبایتی داشته باشد.

مثال

یک فضای آدرس دهی منطقی شامل ۴ صفحه است و هر صفحه حاوی ۲ کلمه است. اگر این صفحات بر روی یک فضای آدرس دهی فیزیکی حاوی ۸ قاب صفحه نگاشت شود، آدرس منطقی و فیزیکی چند بیتی خواهد بود؟

فضای آدرس دهی منطقی: $4 \times 2 = 8 = 2^3$

فضای آدرس دهی فیزیکی: $8 \times 2 = 16 = 2^4$

بنابراین آدرس منطقی ۳ بیتی و آدرس فیزیکی ۴ بیتی است.

مثال

در یک سیستم حافظه صفحه بندی با یک جدول صفحه حاوی ۶۴ مدخل ۱۰ بیتی و صفحه های ۵۱۲ بایتی، آدرس منطقی و فیزیکی چند بیتی است؟

اندازه حافظه منطقی =حاصل ضرب تعداد صفحات (تعداد مدخل ها) در اندازه هر صفحه:

$$64 \times 512 = 2^6 \times 2^9 = 2^{15}$$

اندازه حافظه فیزیکی =حاصل ضرب تعداد آدرسها در اندازه هر صفحه:

$$2^{10} \times 512 = 2^{19}$$

بنابراین آدرس منطقی ۱۵ بیتی و آدرس فیزیکی ۱۹ بیتی است.

مثال

در یک فضای آدرس دهی منطقی هر صفحه حاوی ۵۱۲ کلمه است. صفحات حافظه اصلی بر روی یک فضای آدرس دهی فیزیکی حاوی ۸ قاب صفحه نگاشته می شود. آدرس فیزیکی حاوی چند بیت است؟

$$8 \times 512 = 2^3 \times 2^9 = 2^{12}$$

قطعه بندی

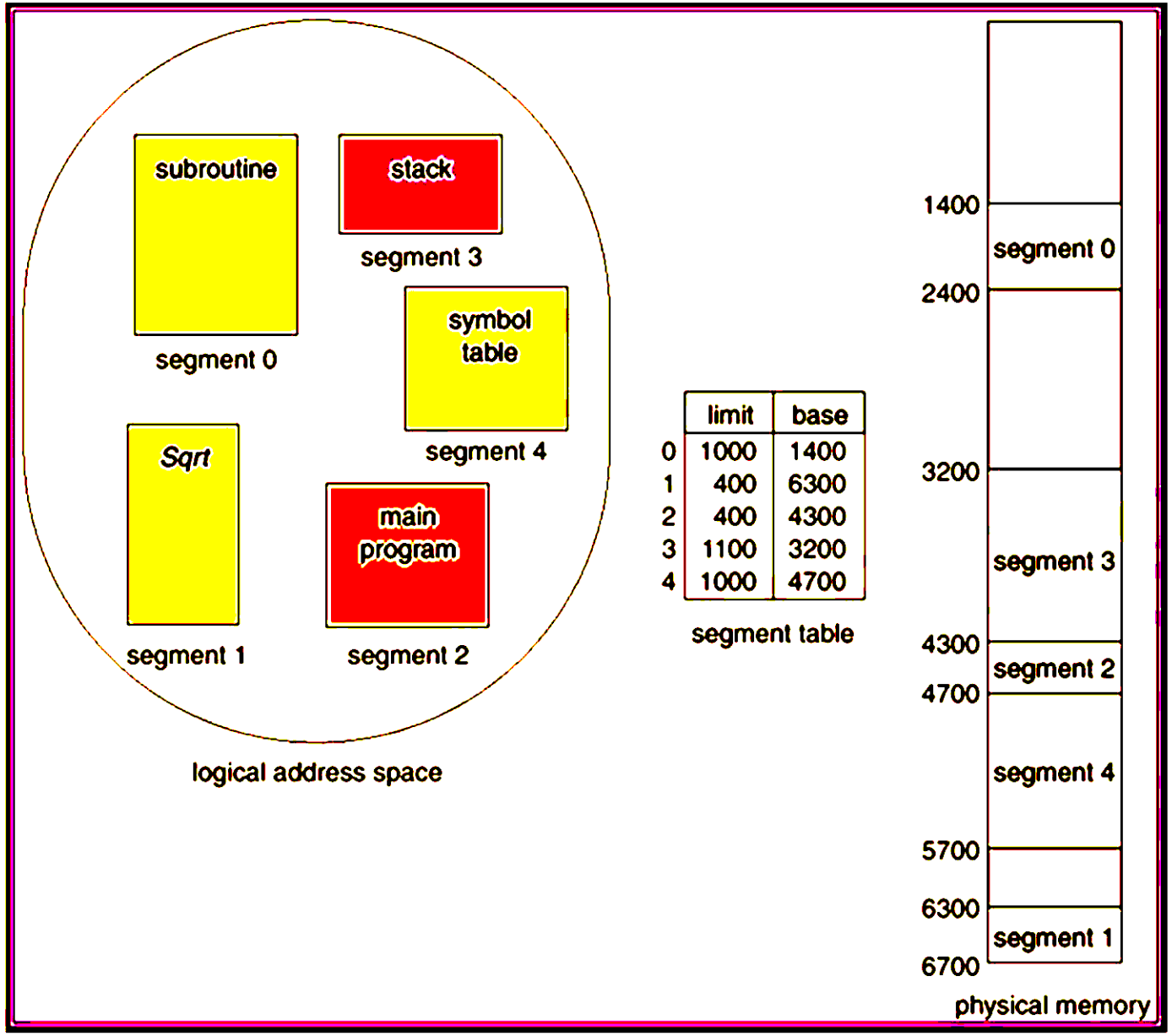
قطعه بندی

در روش قطعه بندی برای مدیریت حافظه، برنامه و داده ها به تعدادی قطعه (Segment) تقسیم می‌شوند و لزومی ندارد اندازه این قطعه‌ها هم اندازه باشند.

هنگامی که یک فرایند به داخل آورده می‌شود، کلیه قطعه‌های آن به داخل حافظه بار می‌شوند و جدول قطعه ایجاد می شود.

هر سطر این جدول شامل آدرس شروع قطعه مورد نظر در حافظه اصلی و طول قطعه می باشد.

دید منطقی از قطعه بندی



نکاتی در رابطه با قطعه بندی

۱- آدرس منطقی از دو قسمت تشکیل یافته است: شماره قطعه و آفست.

۲- امتیاز قطعه بندی، حفاظت از قطعات و اشتراک داده ها و کد می باشد.

۳- الگوی صفحه بندی نمی تواند حافظه فیزیکی را از دیدگاه کاربر نسبت به حافظه تفکیک کند.

۴- قطعه بندی به دلیل بکارگیری قطعه‌های غیرهم اندازه، مشابه بخش بندی پویا است.

البته در قطعه بندی یک برنامه می‌تواند بیش از یک بخش را اشغال کند و لزومی ندارد

این بخشها پیوسته باشند.

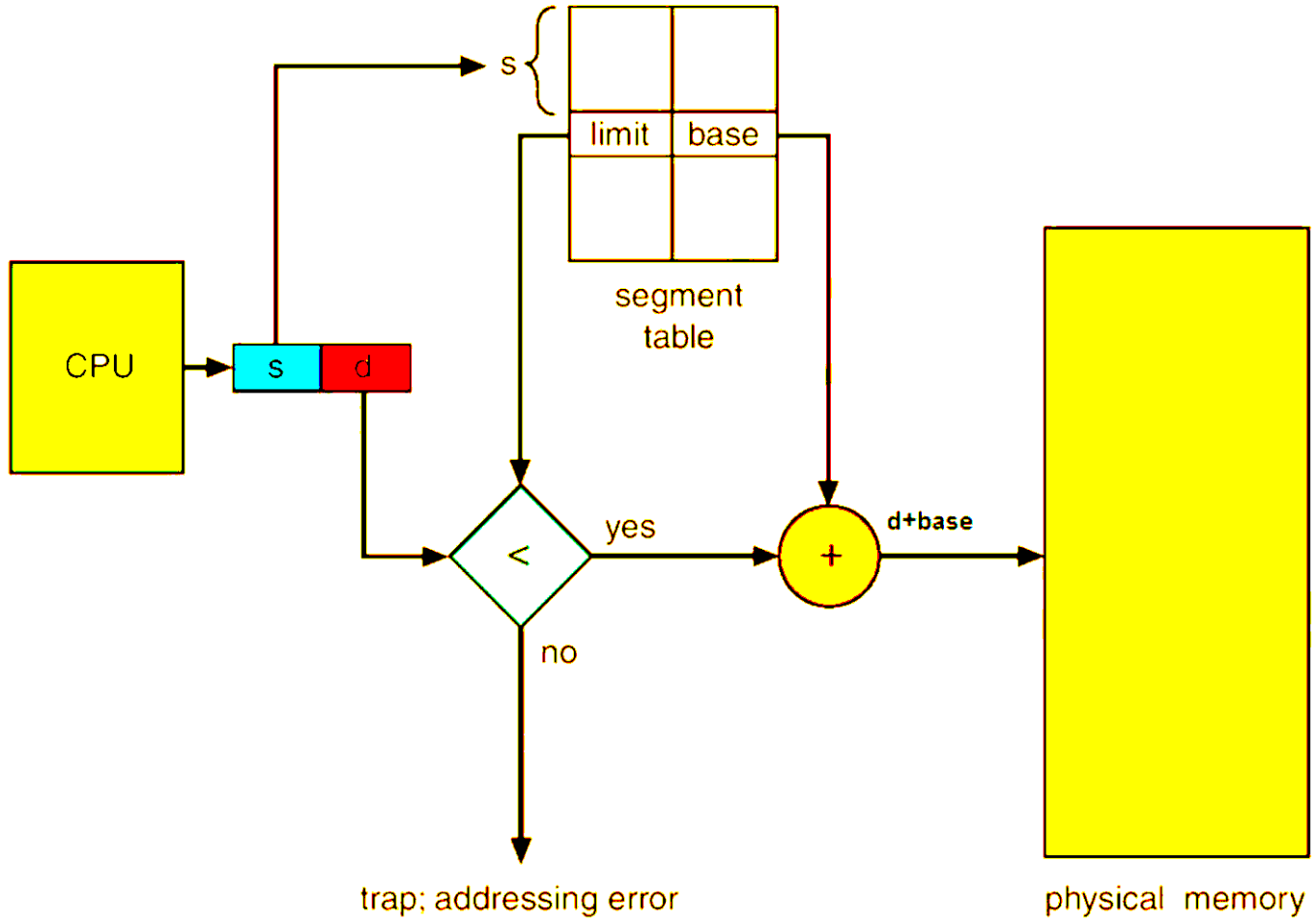
۵- قطعه بندی تکه تکه شدن داخلی را حذف می‌کند، اما دارای تکه تکه شدن خارجی

است.

۶- در حالی که صفحه بندی از دید برنامه‌ساز مخفی است، قطعه بندی قابل رویت و عامل

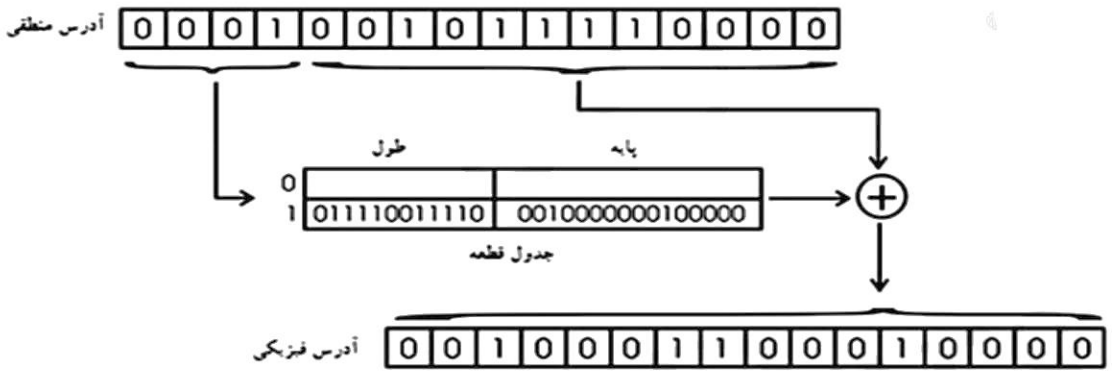
تسهیل سازماندهی برنامه‌ها و داده‌ها می‌باشد.

سخت افزار قطعه بندی



مثال

آدرس منطقی ۱۶ بیتی ۰۰۰۱۰۰۱۰۱۱۱۱۰۰۰۰ مفروض است. نحوه تولید آدرس فیزیکی در روش قطعه بندی را نشان دهید. (آفست ۱۲ بیتی و شماره قطعه ۴ بیتی)



آدرس فیزیکی از جمع آفست ۱۲ بیتی با مقدار پایه بدست می آید.

حداکثر اندازه قطعه : $2^{12} = 4096$

مثال

کدام آدرس منطقی فاقد آدرس فیزیکی هستند؟

2,700

0,150

	Base	Length
0	500	200
1	700	1000
2	400	600

آدرس منطقی در سیستم قطعه بندی از دو قسمت (شماره قطعه و آفست) تشکیل شده،

اولین آدرس منطقی،آدرس فیزیکی ندارد، چون : $700 > 600$

آدرس منطقی دوم ، دارای آدرس فیزیکی است، چون : $150 < 200$

این آدرس برابر است با: $500 + 150 = 650$

پایان