

سیستم های عامل

Operating Systems

میلاذ سلطانی

دانشگاه آزاد اسلامی
واحد تربت جام



فهرست

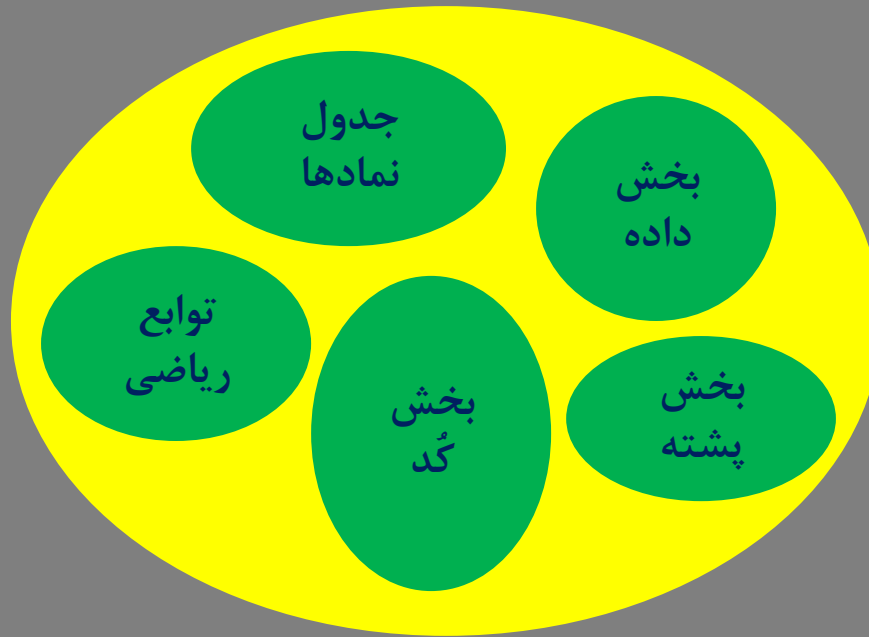
- قطعه بندی (Segmentation)
- بیت های حفاظتی
- ویژگی های سیستم قطعه بندی
- صفحه بندی (Paging)
- تکنیک TLB برای جدول صفحه
- مقایسه صفحه بندی با قطعه بندی
- ترکیب صفحه بندی با قطعه بندی
- مفهوم حافظه مجازی

فصل پنجم

قطعه بندی و صفحه بندی

قطعه بندی (Segmentation)

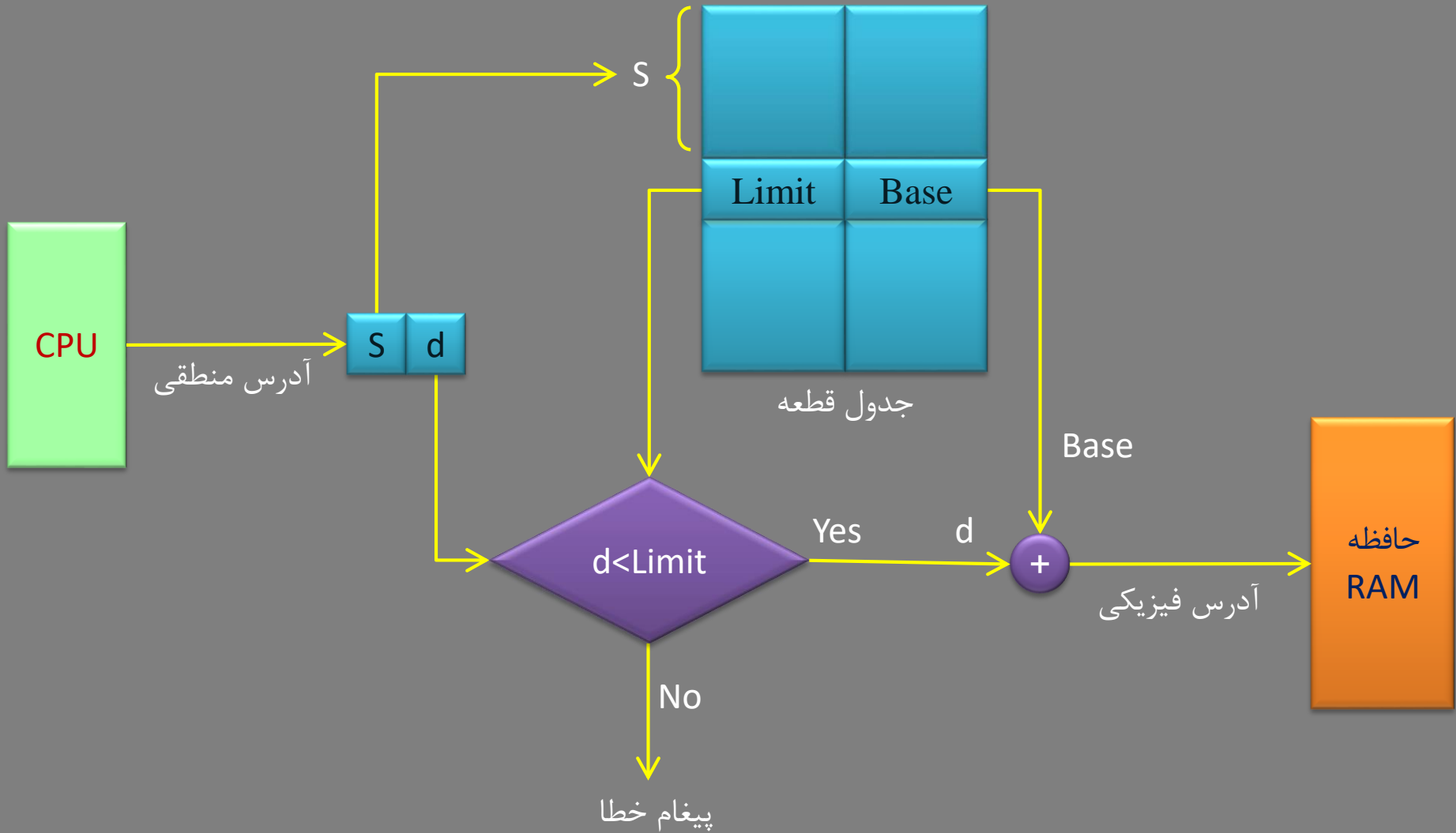
- حافظه از دید فیزیکی آرایه خطی از بایت ها می باشد که هر بایت یک آدرس یکتا دارد.
- کاربر حافظه را از دید منطقی بصورت مجموعه ای از قطعات با اندازه های مختلف می بیند که هیچ ترتیب مشخصی بین این قطعات وجود ندارد.



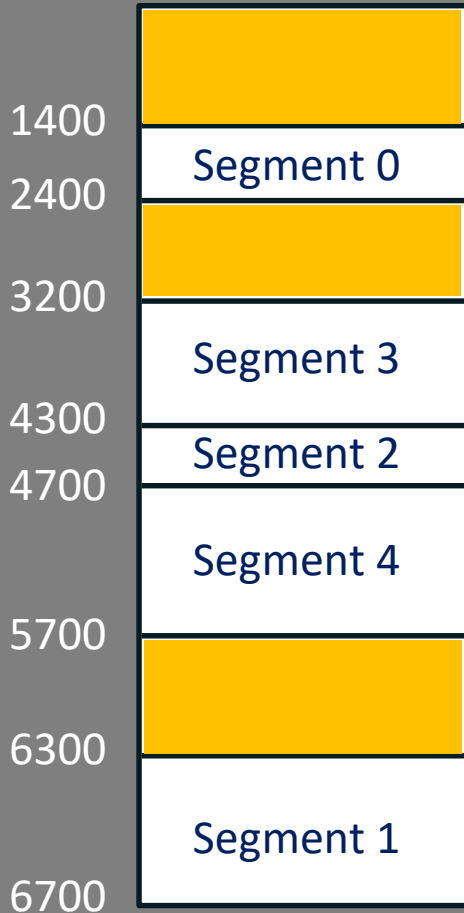
قطعه بندی (Segmentation)

- هر قطعه آدرسی دارد که شامل دو جزء است:
 - آدرس شروع قطعه
 - طول قطعه
- آدرس کاربر در سطح منطقی نیز دو جزء دارد:
 - شماره قطعه
 - فاصله مکان مورد نظر از اول قطعه (آفست)
- تبدیل آدرس منطقی کاربر به آدرس فیزیکی توسط **جدول قطعه** انجام می شود که این جدول از دو ستون تشکیل شده است:
 - « آدرس ابتدای قطعه (Base): یک آدرس فیزیکی در RAM می باشد.
 - « طول قطعه (Limit): محدوده قطعه را مشخص می کند.

تبدیل آدرس منطقی به آدرس فیزیکی در روش قطعه بندی

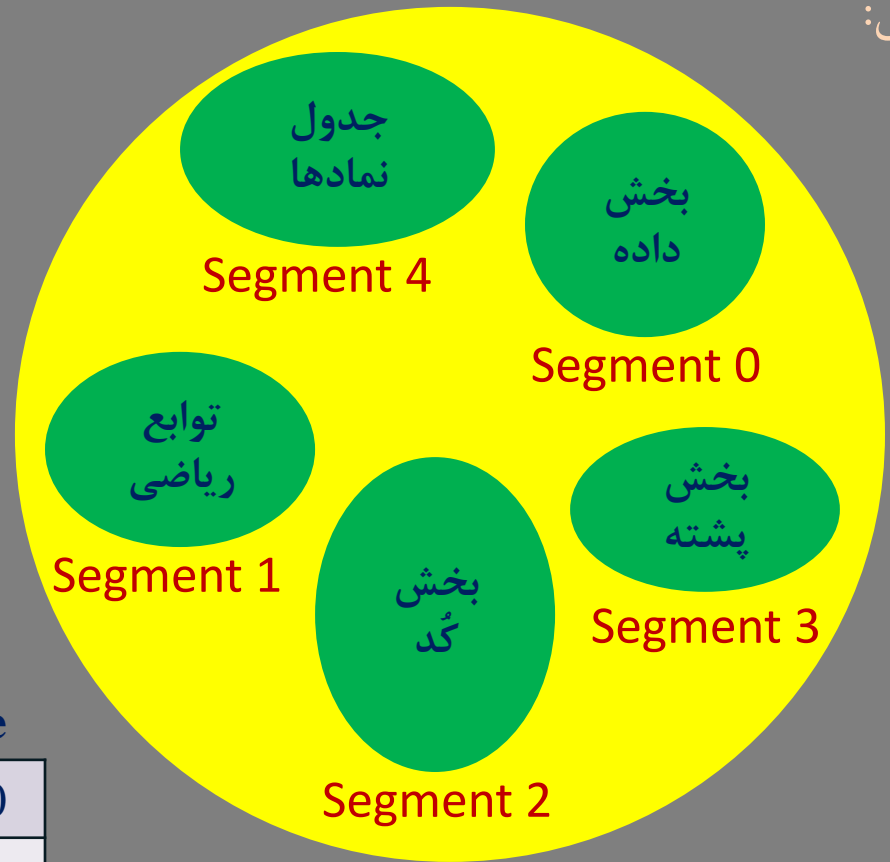


حافظه فیزیکی RAM



| | Limit | Base |
|---|-------|------|
| 0 | 1000 | 1400 |
| 1 | 400 | 6300 |
| 2 | 400 | 4300 |
| 3 | 1100 | 3200 |
| 4 | 1000 | 4700 |

جدول قطعه



فضای آدرس منطقی

بیت های حفاظتی

- در جدول قطعه علاوه بر ستون های **Limit** و **Base** یک سری بیت های کنترلی و حفاظتی وجود دارد. برخی از این بیت ها عبارتند از :
 - **R (Readable)** : این قطعه قابل خواندن است.
 - **W (Writable)** : نوشتن در این قطعه مجاز است.
 - **O (Obey)** : دسترسی به دستورات قطعه و اجرای آنها مجاز است.
 - **A (Accessed)** : هرگاه **CPU** به قطه ای دسترسی پیدا کند این بیت یک می شود.
 - **P (Segment Present)** : مشخص می کند قطعه در حافظه **RAM** هست یا نه !

ویژگی های سیستم قطعه بندی

- مزایا :

- برنامه ها دارای قطعات مختلفی هستند و طول هر کدام می تواند متفاوت باشد، تغییر کند و یا حتی دارای طول صفر باشند.
- تنها قطعاتی از برنامه که به آنها نیاز است وارد حافظه می شوند. به این عمل بار کردن در اثر تقاضا (Demand Loading) گفته می شود.
- تقسیم برنامه به قطعات مختلف منطقی حفاظت بهتری از هر قطعه به عمل می آورد.
- می توان قطعات یک برنامه را بین چند کاربر به اشتراک گذاشت.

- معایب :

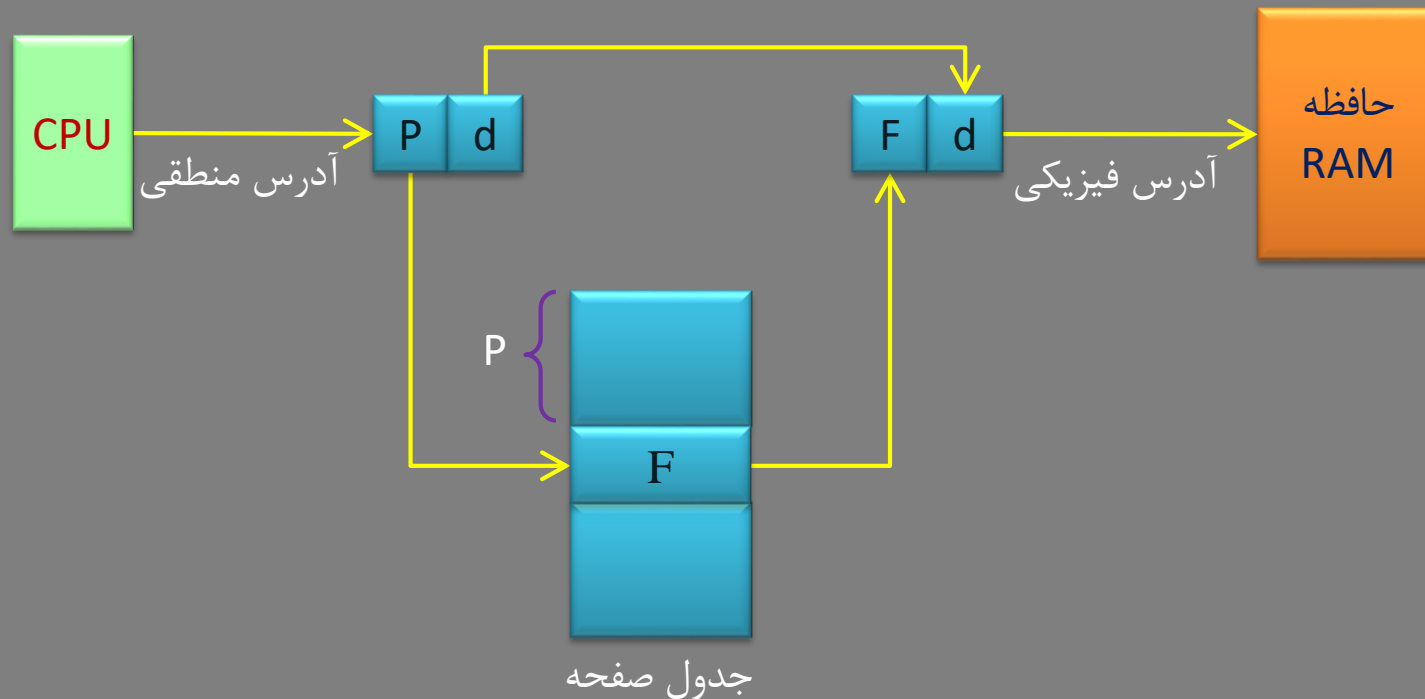
- مشکل اصلی این روش تکه تکه کردن خارجی است.

صفحه بندی (Paging)

- در روش صفحه بندی حافظه فیزیکی به بلوک هایی با اندازه ثابت به نام قاب (Frame) تقسیم می شود. حافظه منطقی نیز به بلوک هایی با اندازه یکسان به نام صفحه (Page) تقسیم می شود.
- اندازه صفحات معمولاً کوچکتر از اندازه قطعات است.
- هنگام درخواست اجرای یک برنامه، صفحات آن برنامه در قاب های حافظه بار می شوند.
- هر آدرس منطقی دارای دو بخش زیر است:
 - شماره صفحه
 - تفاوت مکان از ابتدای صفحه (آفست)

صفحه بندی (Paging)

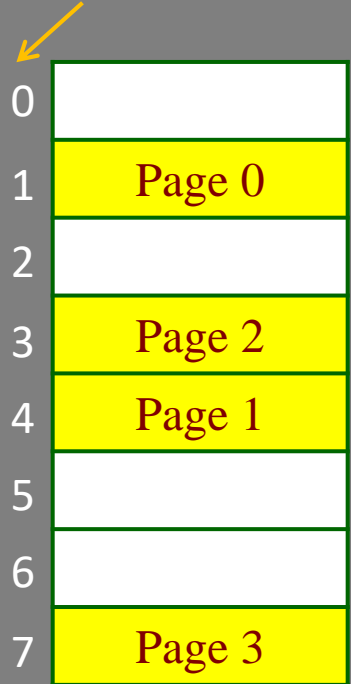
- تبدیل آدرس توسط جدول صفحه که شامل آدرس ابتدای صفحات در حافظه است، انجام می شود.



صفحه بندی (Paging)

- اندازه صفحه و قاب توسط سخت افزار تعیین می شود که این اندازه ها معمولاً توانی از 2 بوده و بین 512 بایت تا 16 مگابایت تغییر می کند.
- همانند جدول قطعه در قطعه بندی، در روش صفحه بندی نیز بیت های حفاظتی وجود دارند.
- اگر یک آدرس منطقی m بیت داشته باشد، آنگاه فضای آدرس منطقی 2^m بایت خواهد بود. اگر اندازه هر صفحه 2^n بایت باشد ($m > n$) در این حالت $(m-n)$ بیت بالاتر فضای آدرس منطقی شماره سطر جدول صفحه (شماره صفحه) و n بیت پایین تر، تفاوت مکان (آفست) می باشد.

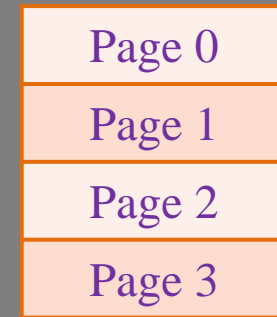
شماره قاب



حافظه فیزیکی RAM

| | |
|---|---|
| 0 | 1 |
| 1 | 4 |
| 2 | 3 |
| 3 | 7 |

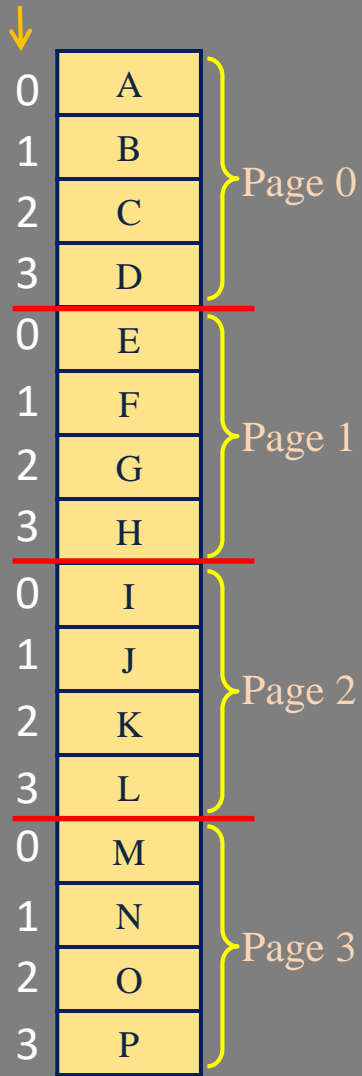
جدول صفحه



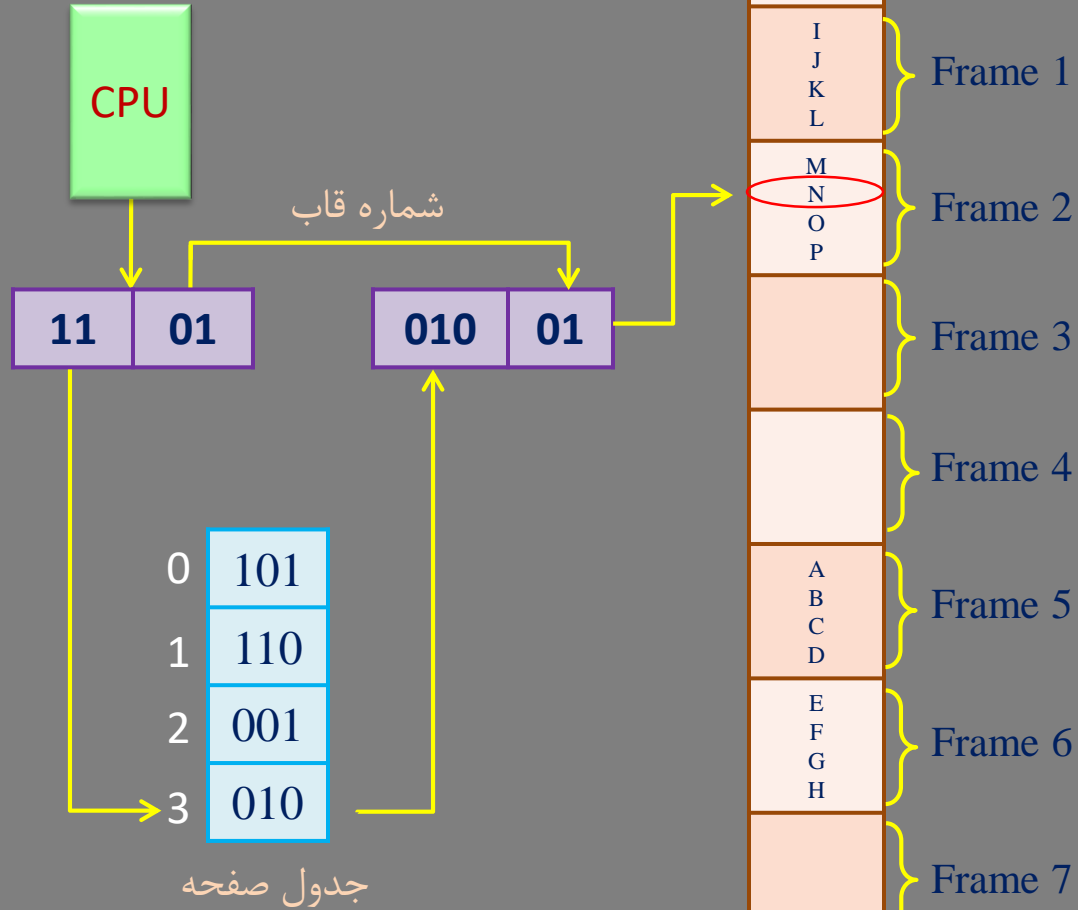
فضای آدرس منطقی

مثال: اگر تعداد صفحات 4 عدد و تعداد قاب ها 8 عدد باشد.

آفست



حافظه منطقی



حافظه فیزیکی

مثال: در یک سیستم اندازه هر صفحه $2^{11} = 2^1 \times 2^{10} = 2 \text{ kB}$ است. اگر در این سیستم آدرس مجازی 32 بیتی باشد، یک برنامه می تواند حداکثر چند صفحه داشته باشد؟

$$\text{حداکثر تعداد صفحات برنامه} = \frac{\text{حداکثر اندازه یک برنامه}}{\text{اندازه هر صفحه}}$$

$$= \frac{2^{32}}{2^{11}} = 2^{21} = 2 \times 2^{20} = 2 \text{ M}$$

در این سیستم یک برنامه با این مشخصات می تواند تقریباً دارای دو میلیون صفحه باشد.

ویژگی های صفحه بندی (Paging)

- امکان تخصیص حافظه بصورت غیر همجوار وجود دارد.
- تکه تکه شدن خارجی وجود ندارد.
- تکه تکه شدن داخلی وجود دارد.

نکته ۱ : اگر برای کاهش تکه تکه شدن داخلی اندازه صفحات را کم کنیم، تعداد صفحات بیشتر می شود. لذا اندازه جدول صفحه و زمان تبدیل آدرس افزایش می یابد.

نکته ۲ : در مدل صفحه بندی ساده اندازه حافظه فیزیکی باید بزرگتر یا برابر فضای آدرس منطقی باشد.

تکنیک TLB برای جدول صفحه

- ثبات های انجمنی یا بافرهای کناری (Translation Look-aside Buffers) از حافظه های با سرعت بسیار بالا ساخته شده اند.
- این ثبات ها دارای دو بخش کلید و مقدار هستند.
- وقتی شماره صفحه به این ثبات داده می شود، با تمام کلیدها بصورت همزمان مقایسه خواهد شد.

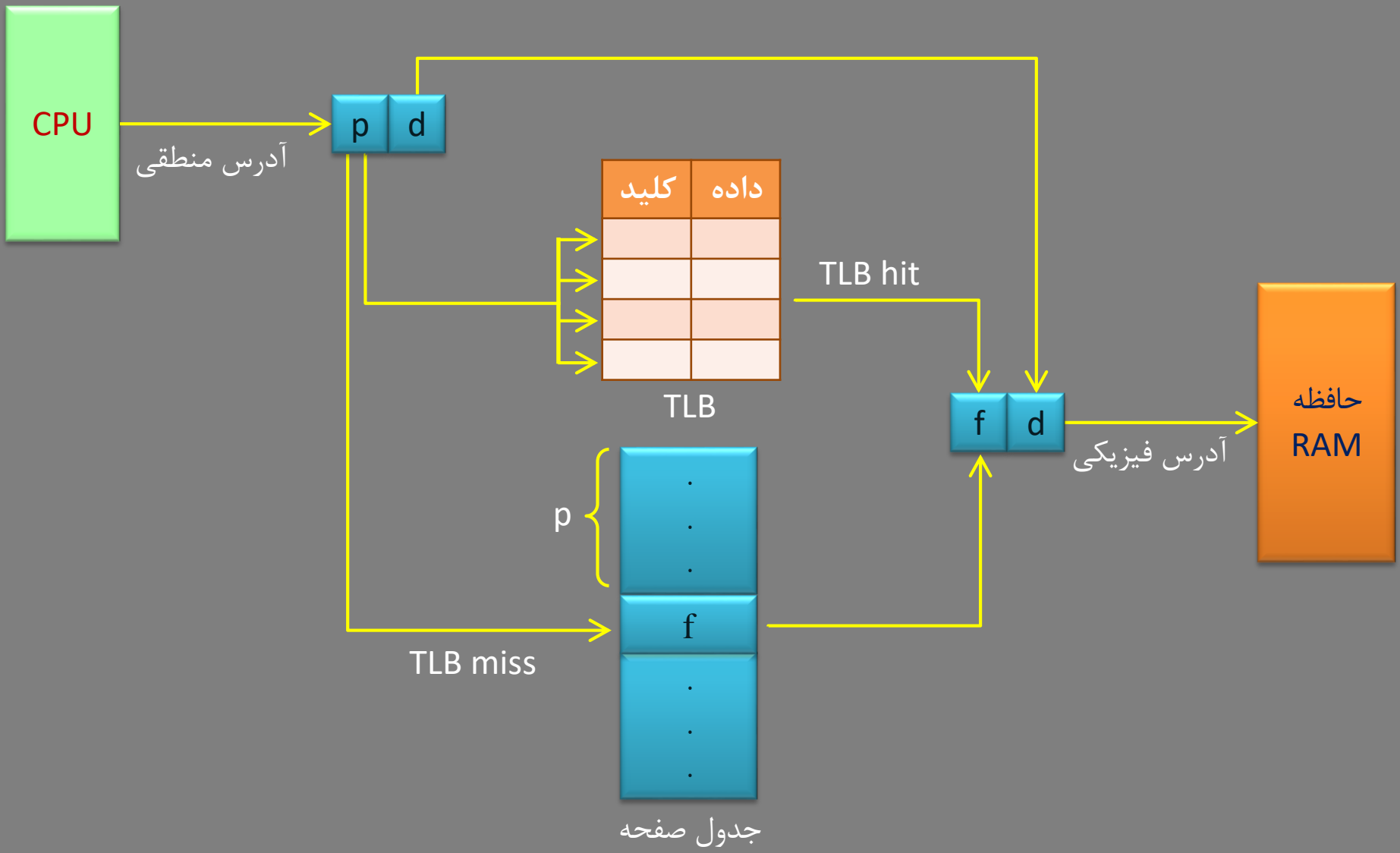
| کلید (TAG) | مقدار (DATA) |
|------------|--------------|
| 6 | 18 |
| 5 | 9 |
| 3 | 0 |
| . | . |
| . | . |
| . | . |
| 7 | 12 |

5

9

- این ثبات ها بسیار گران بوده و همواره ترکیبی از جدول صفحه و ثبات های TLB استفاده می شود.
- تعداد کلید های آن معمولاً 8 تا 2048 عدد می باشد.

استفاده از تکنیک TLB در صفحه بندی



مثال: اگر جستجوی TLB به اندازه ۲۰ نانو ثانیه طول بکشد و دستیابی به حافظه اصلی نیز ۱۰۰ نانو ثانیه زمان بخواهد، آنگاه زمان دستیابی به حافظه را در مواقعی که شماره قاب در TLB پیدا شود و هنگامی که در TLB پیدا نشود را محاسبه کنید

اگر شماره قاب در TLB پیدا شود :

زمان دستیابی به حافظه اصلی + زمان جستجوی TLB = زمان دسترسی به حافظه

$$20 + 100 = 120 \text{ nS}$$

اگر شماره قاب در TLB پیدا نشود :

(۲ × زمان دستیابی به حافظه اصلی) + زمان جستجوی TLB = زمان دسترسی به حافظه

$$20 + 100 \times 2 = 220 \text{ nS}$$

مقایسه قطعه بندی با صفحه بندی

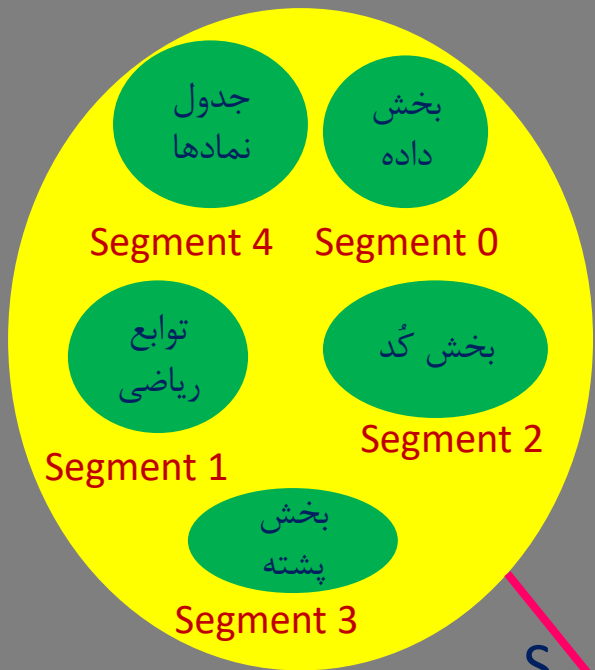
| موارد مقایسه | صفحه بندی | قطعه بندی |
|--|--|---|
| آیا برنامه نویس از بکارگیری این تکنیک آگاه است؟ | خیر | بله |
| آیا فضای آدرس دهی منطقی می تواند از اندازه حافظه فیزیکی بیشتر باشد؟ | بله | بله |
| آیا کد برنامه و داده ها می توانند جدا از هم بوده و بصورت متفاوت محافظت شوند؟ | خیر | بله |
| آیا به اشتراک گذاشتن کد برنامه بین کاربران به سادگی امکان پذیر است؟ | خیر | بله |
| علت اصلی ابداع تکنیک چه بوده است؟ | داشتن فضای منطقی بزرگتر از فضای فیزیکی (حافظه مجازی) | امکان جداسازی منطقی اجزای برنامه از نظر عملکرد و استفاده اشتراکی و حفاظت بهتر |
| تکه تکه شدن داخلی | دارد | ندارد |
| تکه تکه شدن خارجی | ندارد | دارد |

ترکیب قطعه بندی با صفحه بندی

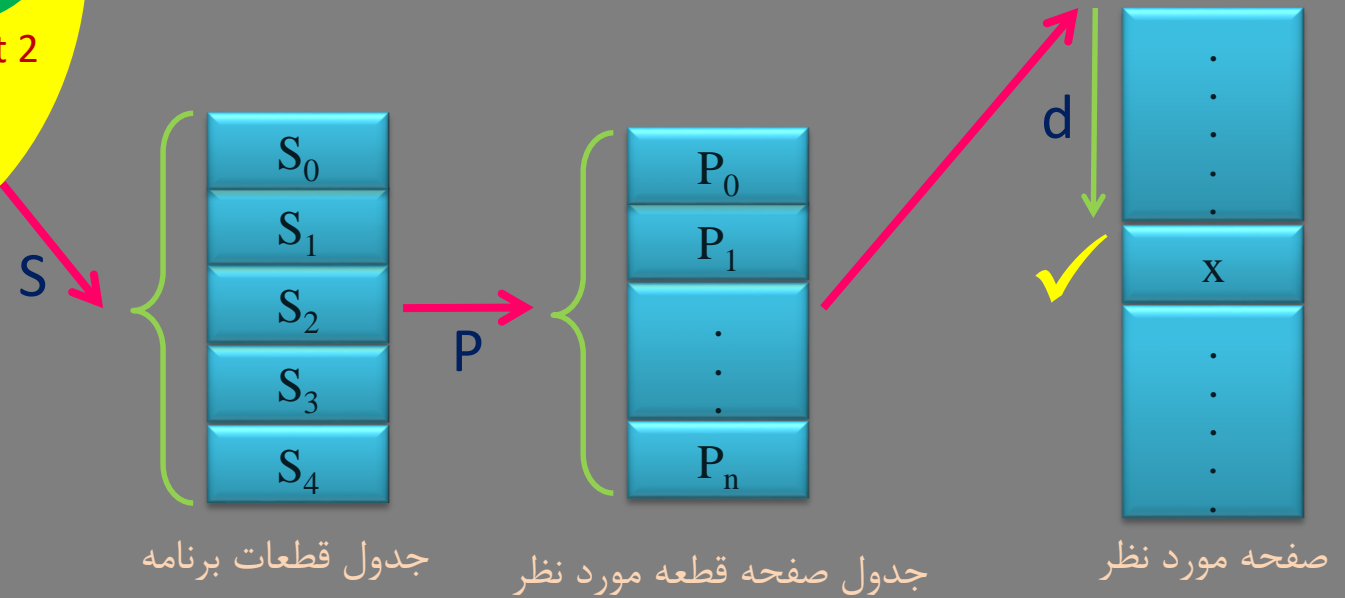
- در سیستم های امروزی از ترکیب دو روش قطعه بندی و صفحه بندی استفاده می شود. یعنی ابتدا برنامه قطعه بندی می شود، سپس هر قطعه صفحه بندی می گردد.
- چنین سیستمی هم مزایای قطعه بندی (حافظت و امنیت) و هم مزایای صفحه بندی (حافظه مجازی و عدم تکه تکه شدن خارجی) را دارد.
- در این سیستم ها آدرس منطقی دارای ساختار زیر است :

| شماره قطعه Segment Number | شماره صفحه Page Number | تفاوت مکان Displacement |
|------------------------------|---------------------------|----------------------------|
|------------------------------|---------------------------|----------------------------|

تبدیل آدرس در روش ترکیبی قطعه بندی و صفحه بندی



فضای آدرس منطقی
برای یک برنامه



جدول قطعات برنامه

جدول صفحه قطعه مورد نظر

صفحه مورد نظر

$$\text{آدرس مورد نظر} = \boxed{\text{تفاوت مکان } d} \quad \boxed{\text{شماره صفحه } P} \quad \boxed{\text{شماره قطعه } S}$$

مفهوم حافظه مجازی

- در حافظه مجازی فرآیندها بدون اینکه کاملاً در حافظه قرار گیرند، می توانند اجرا شوند. یعنی اندازه برنامه ها (حافظه منطقی) می تواند بزرگتر از اندازه حافظه (فیزیکی) باشد.
- مزایای روش حافظه مجازی عبارتند از :
 - چون برنامه ها حافظه فیزیکی کمتری اشغال می کنند، برنامه های بیشتری همزمان قابل اجرا بوده و بهره وری (راندمان) CPU را افزایش می دهد.
 - برنامه ها می توانند هر اندازه باشند. لذا برنامه سازی برای برنامه نویسان ساده تر می شود.
 - عملیات I/O کمتری جهت اجرای برنامه نیاز است، لذا سرعت اجرای برنامه افزایش می یابد.

مفهوم حافظه مجازی

- مفهوم حافظه مجازی در سیستم های امروزی اغلب همراه با روش صفحه بندی نیازی (Demand Paging) استفاده می شود، ولی باید توجه داشت که این دو روش از هم جدا هستند.

| | |
|---|---|
| 0 | A |
| 1 | B |
| 2 | C |
| 3 | D |
| 4 | E |
| 5 | F |

فضای حافظه منطقی
(برنامه در دیسک)

| | | |
|---|---|---|
| 0 | 1 | V |
| 1 | | i |
| 2 | 0 | V |
| 3 | 2 | V |
| 4 | | i |
| 5 | 3 | V |

جدول صفحه

| | |
|---|---|
| 0 | C |
| 1 | A |
| 2 | D |
| 3 | F |

حافظه فیزیکی RAM

سؤال ؟